ARPA ORDER NO.: 189-1

R-656-ARPA

December 1971

PREMIR: A Prediction Model for Infiltration Routes

C. F. Black and L. J. Pipes

A Report prepared for

ADVANCED RESEARCH PROJECTS AGENCY

Rand
SANTA MONICA, CA. 90406

This research is supported by the Advanced Research Projects Agency under Contract No. DAHC15 67 C 0142. Views or conclusions contained in this study should not be interpreted as representing the official opinion or policy of Rand or of ARPA.

ARPA ORDER NO.: 189-1

R-656-ARPA

December 1971

PREMIR: A Prediction Model for Infiltration Routes

C. F. Black and L. J. Pipes

A Report prepared for

ADVANCED RESEARCH PROJECTS AGENCY

Rand

SANTA MONICA, CA. 90406

Published by The Rand Corporation

PREFACE

This Report is a product of an investigation of border infiltration undertaken by The Rand Corporation for the Advanced Research Projects Agency. It completely discusses the logic underlying PREMIR (a Prediction Model for Infiltration Routes) and provides information on the model and illustrative results from its first application. Most studies of infiltration deal with the problem in the context of detecting enemy logistics and operations through aerial reconnaissance, prisoner or informer interrogation, reports from the field, etc. These studies have generally not been concerned with explicitly developing the relationships between patterns of infiltration routes and the infiltrators' criteria for choosing them. In sharp contrast, the present inquiry is intended to more fully identify and formulate the essentials under which infiltration routes may be systematically analyzed, modeled, and predicted.

Although the analytical techniques employed in this study are well established, the particular application of them in an investigation of border infiltration is believed novel. For that reason, the information presented here should be of prime concern to others interested in the problem of infiltration. But as the study illuminates certain aspects of infiltration that have so far received little attention in the literature, it should also be of interest to a much wider audience.

SUMMARY

This Report investigates some conditions under which infiltration routes can be systematically modeled, analyzed, and predicted, and identifies some areas where the resulting information may be efficiently utilized. PREMIR (Prediction Model for Infiltration Routes) is primarily a mathematical computer model which by means of established analytical techniques arrives at a set of predicted (optimal) infiltration routes between various start and specific end points within a particular area of study.

The ability to model real infiltration routes rests heavily on the assumption that there is a rationale underlying the choice of such routes, the criterion (or criteria) for which can be discovered by the modeler. The criterion usually involves the minimization of some costs, and where these costs can be expressed quantitatively, the techniques of dynamic programming can be applied to determine minimum-cost paths.

An inefficient infiltration route may incur many additional costs for the infiltrator, such as having to spend more time on the route, carry more food and water, endure more physical and psychological stress, and undergo greater exposure to a variety of risks. The model presented here expresses many of these costs in terms of caloric requirements, and the resulting predicted routes represent a minimization of this metric.

Although we experimented with cost functions that included both a linear and a quadratic dependence on grade in order to allow for the lower energy requirement of small negative grades, the results were not entirely satisfactory because of uncertainty as to the relative weights of the linear and quadratic terms. The usefulness of a linear term in the cost function is further nullified because between the same start and end points a linear term contributes the same cost to all routes, since the total cost due to this term depends only on the altitude difference between the start and end points.

Since infiltration routes are traveled in both directions (because of personnel rotation and transportation of supplies from either direction), and since an antisymmetric linear term does not contribute to

route differentiation, the final form of the cost function employed in the model is symmetric in the cost associated with the altitude change along a predicted route (i.e., the cost associated with that part of the route which goes downhill is the same as the cost associated with that part going uphill, traveling in the same direction for equal distances). This more useful form is

$$\text{Cost} = 1 + (\alpha \, |\text{grade}|)^{p}.$$

Values for p of 1 and 2 were investigated. Values investigated for α were 10, 20, and 40.

The model presently uses only terrain data in its cost function but is easily expanded to include other data as they become available. Among the necessary inputs to the model are altitude readings and the specification of the terminal point, all within the area under study. A two-dimensional interpolation function may be employed to interpolate for altitude values between the input grid data points, resulting in a finer network representation of the study area. The minimum-cost paths are computed from all included points.

Since the optimal path *from* any point is unique and independent of the path *to* that point, it is sufficient for the computer to print out a table that specifies to which of the eight adjacent points one proceeds from any point on the grid en route to the terminal. Also obtained at each point is a cost measure representing the total computed cost from each point to the terminal.

The model has been applied to a rectangular-shaped area, 210 × 220 km. This grid covers part of northern South Vietnam and adjacent Laos. The road network in Laos was used as the location for start points for determining infiltration routes which terminated in various assumed base-camp areas in South Vietnam. The results obtained from this first application resembled many characteristics of observed infiltration routes and offered an explanation for some route characteristics not previously understood.

The model is potentially widely applicable. The main areas of application are as follows:

1. *Infiltration route characteristics.* It has never been seriously

suggested that infiltration routes are chosen at random. Traditional communication routes almost always follow easy rather than difficult terrain. The model provides a systematic approach for investigating the criteria and degree of randomness involved in route determination, provided some infiltration routes are known to the investigator. The model can be useful in indicating probable infiltration routes and patterns; identifying likely alternative routes (for example, after the user has blocked optimal routes); and estimating logistics and time requirements for infiltration.

If there were some systematic departure of the real route from the "terrain optimal" routes in some region, an investigation would then be made as to what conditions in the region caused the routes to differ. Perhaps the existence of a village, a traditionally established path, vegetation cover, or unknown base camps could explain the deviation. Once a reason for the deviation is conjectured, a new cost function can be constructed which includes a weighting factor for the new condition(s).

Route characteristics other than the optimizing criteria can be inferred. For example, it may be possible to discover camps or rallying points by examining the deviation of real routes from those predicted. Or it may be possible to determine start or end points for the real routes by matching fragmentary route information to routes predicted using an array of start and end points.

2. *Barrier and sensor emplacement.* The model provides a useful indicator of the best places for locating barriers, sensors, or interdiction campaigns and allows for the study of cost sensitivity to changes in routes after blocking or detection.

3. *Offensive applications.* The model should be of value to the infiltrator as well as the defender, especially where the infiltrating force has little experience.

The model can also be applied in the areas of selecting routes for new roads or for ground resupply; finding ambush points; or determining boundaries of political subdivisions on the basis of ease of communication--an area which is too broad for further consideration at this time. The model is conceptually simple, but it is diverse in the parameters and implications it encompasses and therefore provides a rich field for further investigation.

CONTENTS

PREFACE iii

SUMMARY v

Section
I. INTRODUCTION 1
Background 1
Factors Influencing Choice of Infiltration Routes 3

II. DATA COLLECTION AND INTERPOLATION 9

III. THE COST EQUATION 15

IV. THE COMPUTER MODEL 23
Main Program 24
Interpolation 25
Cost Calculation 25
Dynamic Programming 27

V. THE VIETNAM EXAMPLE 31

VI. APPLICATIONS OF THE MODEL 45
Infiltration Route Characteristics 45
Barrier and Sensor Emplacement 46
Offense Applications 47

Appendix
A. A JOSS PROGRAM FOR INVERSE-SQUARE INTERPOLATION 49
B. PREPARATION OF AN INPUT DECK FOR THE FORTRAN IV VERSION OF PREMIR 53
C. FLOW CHARTS OF THE PRINCIPAL SUBROUTINES TO PREMIR 59
D. FORTRAN IV SOURCE LISTING OF PREMIR 69

REFERENCES 77

R-656
/0

I. INTRODUCTION

BACKGROUND

Many of the political boundaries separating long-established countries of the world lie in rugged terrain. The barrier that such terrain imposed on communications was often responsible for the cultural and political differentiation. Today, when warfare occurs between countries where such barriers exist, it may maintain many primitive aspects, particularly with regard to logistics and communications across these boundaries, in spite of modern technological developments provided by new, larger powers. Because modern implements of warfare such as aircraft and tanks cannot operate effectively over rugged terrain covered with thick vegetation, insurgent groups fighting civil wars often use such terrain for their base camps and hidden supply routes. Therefore, transportation and communication through rough terrain on foot or by small animal or bicycle pack remains an important aspect of modern warfare at the lower end of the technological spectrum.

An initial model for simulating, analyzing, and predicting infiltration routes has been completed. PREMIR (Prediction Model for Infiltration Routes) is primarily a mathematical computer model which by means of established analytical techniques arrives at a set of predicted (optimal) infiltration routes between various start and specific end points within a particular area of study. Although the model presently uses only terrain data in its cost function, it can be easily expanded to include other data as they become available.

Travel in rough terrain is more difficult than is generally thought by those who have little experience in such travel. As a rule of thumb, it requires 10 times more energy to ascend 1 foot in altitude than to cover 1 foot on horizontal terrain.* This difference in energy requirements becomes even greater when a heavily laden bicycle is being pushed up an incline rather than over flat land. Consequently, all possible infiltration routes through rugged terrain are not equally suitable and, more important, are not equally costly.

---

*Or twice as much energy to ascend a 10 percent grade as to cover a horizontal path.

R-656
11

Of course, terrain alone is not the only important factor in choosing infiltration routes. The availability of vegetation to hide the routes, the availability of food and water, the avoidance of unfriendly population, and the availability of friendly population will also alter an infiltration route based upon terrain alone.

The idea of an infiltration route assumes that there is a definite start point and end point. However, in the course of a conflict, these points may change and the corresponding routes may change, so that one must be careful in assuming that such routes have a permanent character or that the start points and end points will be known in advance. Nevertheless, in order to develop a model for predicting infiltration routes, it is necessary to make certain idealizing assumptions, and one of the first assumptions usually employed is that terminal points or initial points are known. Later on we will be able to relax even this assumption.

The ability to model real infiltration routes rests on the assumption that there is a rationale behind choosing such routes, the criteria for which can be discovered by the modeler. In other words, infiltration routes are not chosen completely at random but are chosen to minimize some costs, whether they be physical, economic, or psychological. In the model we have developed, it is assumed that these cost factors can be measured by a common metric. For example, one might be willing to walk 10 miles out of his way to avoid scaling a 1-mile-high peak, or to enable 50 miles of his path to lie beneath a jungle canopy rather than in the open, or to avoid passing within 1 mile of an unfriendly village.

With such cost factors based on a common metric it is possible to utilize a dynamic-programming algorithm which will determine the minimum-cost path between two points. These predicted paths can be used in many ways. For example, one can assume a start point such as a road in Laos and an end point such as a base camp in South Vietnam and find the minimum-cost infiltration route based on the assumed form of the cost function. If segments of an infiltration route are already known through aerial reconnaissance or prisoner interrogations, these segments can be compared to the prediction, and differences can be resolved by trying

new cost functions or by trying new start and end points. In this manner, we have a systematic way of investigating the rationale used by the infiltrator in choosing his infiltration routes.

The model can also be used to locate optimal points for constructing infiltration barriers or for interdicting infiltration. Barriers can be introduced into the model mathematically by assigning high costs to certain points across an infiltration route and observing the new infiltration routes predicted by the model which circumvent these barriers. A comparison of the costs to the infiltrator before and after barrier installation gives a measure of the effectiveness of the specific barrier installation. One would expect barriers to be most effective at choke points, that is, regions where many infiltration routes with different start and end points pass close to one another because of the high cost associated with alternate areas. These choke points show up quite clearly in some of the model calculations that have been performed.

The area from which terrain data were gathered included the northern portion of South Vietnam and the adjacent areas of Laos. The road network in Laos was used as the location for start points for determining infiltration routes which terminated in various base-camp areas assumed in Vietnam. The results obtained resembled many characteristics of observed infiltration routes and offered an explanation for some of the route characteristics not previously understood.

FACTORS INFLUENCING CHOICE OF INFILTRATION ROUTES

An inefficient infiltration route may incur many costs associated with factors such as the following: The infiltrator may have to spend more time en route; carry more food or water; endure more physical and psychological stress; risk greater exposure to disease as a result of insect or animal bites; risk greater chance of discovery by the enemy; and risk greater chance of losing his way and not reaching his destination.

The food and water requirement may be the most convenient means of measuring the overall cost factors associated with physical stress. In many parts of the world, springs and streams are readily available,

so that water need not be carried by the infiltrator except in small amounts. Food, on the other hand, is not often available in rugged terrain and is usually carried by the infiltrator; in Southeast Asia rice is most commonly carried. In Indochina, the infiltrator, who may be on foot for as long as 3 to 6 months, generally cannot carry with him all the food that will be required for such a trip. Therefore, logistics bases and rest areas will usually be set up along the infiltration route, giving the route a certain characteristic of permanence, since it is not easy to change the logistics system frequently.

The cost of this logistics system itself will be proportional to the rate at which food is consumed by individual infiltrators and to the total number of infiltrators. Since the food will usually be brought into the supply areas by some primitive methods, the rough terrain imposes an additional burden on the logistics system as well as on the infiltrator. In fact, the choice of infiltration routes may be constrained more by the need for transportation of supplies to the rest areas along the trail than by the stress placed on the infiltrator as an individual.

Since supplies may have to move in a direction opposite to that of infiltration, such as from the rice paddies of Vietnam up into the highlands, we cannot assume that costs should be assigned only to that part of the infiltration route which is downhill for the infiltrator. For this reason, in our final form of the cost function, we have assumed symmetry in the cost associated with altitude change along an infiltration route; that is, the cost of that part of the route which goes downhill is as great as the cost of that part going uphill. This form of the function, however, is not final and may be changed if evidence indicates it should be otherwise.

For the reasons discussed, we have assumed that the amount of food required per infiltrator will be an important factor in determining the infiltration routes employed. The rate at which food is consumed will depend upon a number of variables which we will discuss.

We assume that food is consumed primarily for calories, other nutrients being obtained in the process with little additional cost. The

calories required per mile of travel will depend upon the speed of travel. An infiltrator running along flat land consumes more calories per mile than one traveling more slowly at a comfortable speed of 2 or 3 miles per hour. Traveling too slowly also requires greater amounts of food per mile. Traveling up a slope requires more calories than traversing flat land. Data provided later in this Report indicate that 10 times as many calories are required for every foot of altitude gained as are required for 1 foot of horizontal travel.

The roughness of the terrain will also be a factor in determining the difficulty of the path and hence the amount of food required. It is easier to walk a smooth path on flat land than it is to walk along a ridgeline or along the bottom of a narrow valley, even though no net change in altitude is involved. Furthermore, ridgelines and valleys are often easier to follow than paths following the contour lines on steep hillsides, since one must exert additional energy to obtain solid footing and to avoid slipping down the hill. Even if the paths are well prepared, erosion will often remove level surfaces from sections of the path.

Temperature, humidity, and altitude have also been found to affect calorie requirements. When temperatures are very low, additional food is required to maintain body temperature; when temperatures are very high, metabolic efficiency is reduced and additional calories are required to do the same work done in moderate temperatures. High humidity causes reduction in efficiency of heat loss from the body and a consequent rise in body temperature to an inefficient level. During conditions of very low humidity, body-water loss is increased, and additional amounts of water may have to be carried to replace this loss. The metabolic efficiency of a person not adapted to high altitudes may be reduced when he must function in such areas.

All of these factors are augmented if additional weight must be carried by the infiltrator. For example, if there is a long distance between logistics bases, a considerable amount of additional food and water may have to be carried, and in turn more food and water is required to carry the additional weight, so that for long distances the weight to be carried increases more than linearly with the distance

traversed. The problem becomes even more severe if the weight carried is increased by weapons and ammunition.

It would appear that establishing infiltration routes through the jungle is less of a problem than was first thought. Most jungles are heavily populated with animals, which have developed trails for their own movement through the jungle. Some of these trails cover considerable distance--often in favorable directions for the infiltrators--as seasonal factors require changes in feeding habits. Furthermore, in situations such as that in South Vietnam, where infiltration routes have been developed over years and tens of thousands of personnel have traversed these routes, the initial cost of clearing routes that are optimal on the basis of other considerations is small.

The jungle can actually be a preferred area for infiltration routes because of the cover it provides against detection from the air. However, in Vietnam where jungle covers are plentiful, many infiltration routes are found to pass through open areas. This may imply that the need for air cover is not an important determinant in the selection of infiltration routes.

The presence of friendly or unfriendly villages may alter the course chosen for an infiltration route. Friendly villages can serve as rest and resupply areas for the infiltrator and as a source of food and water. Establishment of infiltration routes near unfriendly villages can result in discovery of these routes and would thus be avoided by those establishing them. However, it is probably important to consider that in the course of civil war, the loyalties of various villages may shift from one side to the other, and even a friendly village may have an informer; thus it would perhaps be in the best interest of the infiltrator to establish his important routes far from any population centers. Such a choice may be very difficult in some cases, however, since villages in mountainous areas will probably be connected by long-established communication routes which may very well be the best and most easily passable in the area, particularly if the terrain is rugged.

A river can be either a barrier or an aid to infiltration, depending upon its type of flow and direction relative to the direction of

infiltration. In the lowlands, rivers tend to be broad and slow-moving, and they leave flat valleys which are easily followed. However, in much of the world and particularly in Southeast Asia, these river valleys (because of their fertility and the ease of communication offered by the river) are often heavily populated, so that use of the river by the infiltrator involves passing through populated areas, with the implications described above.

In mountainous country, rivers tend to be narrow and swift, and the passages cut through the mountains are often narrow, with no adjacent footpaths along the shore. In Southeast Asia, the rivers do not seem to form difficult barriers for most infiltration routes, since they are easily crossed by rope suspension bridges or by improvised ferry service. Furthermore, a swift river may be an advantage in that it often provides a good source of fresh water.

Seasonal factors may play an important role in route determination. Routes which are very good during one season of the year may become impassable during others. During the rainy season in Southeast Asia, for example, some infiltration routes are completely covered by flood waters; others may become so thick with mud that it is impossible to move bicycles along them, and walking requires additional energy consumption. Rain may also seriously interfere with the transportation of food such as rice, which deteriorates in moisture. Serious hazards are presented in desert areas during the infrequent but heavy rains. Because of the bareness of the hillsides, water collects very rapidly in arroyos and may move as a wall several feet high down narrow gulleys at a high rate of speed. Passage through narrow canyons in the desert during the rainy season can therefore be very hazardous, and infiltration routes at higher altitudes might be preferred.

During the summer season passage through desert areas may become difficult due to the extreme heat, requiring the infiltrators to move only at night. Traffic at night, however, must be accomplished in complete darkness, since infiltrators are readily spotted at great distances if they are using any light to illuminate their path. Consequently, infiltration routes for night travel through desert areas must be chosen with more caution. A greater cost would therefore be assigned to difficult mountainous routes.

The unavailability of water is always a serious problem in the desert, especially during the summer season, since considerable quantities of water may be required to replace the high water loss of the body. Additional water must therefore be carried and thus a greater cost is incurred. One would expect infiltration routes to be altered to take advantage of any water availability along the way.

In winter different kinds of problems are presented to the infiltrator. Extreme cold may require that he build fires to prepare food and to keep warm during the night. In such cases, covered terrain will be more desirable in order that the fires will not be readily detected. The cover may be thick vegetation or caves and rocky shelters. Heavy snowfall may make mountain canyons impassable for certain kinds of traffic. On the other hand, in ski country, some passages may be rendered considerably easier by the presence of snow, troops being able to cover many more miles per day on skis than they could on foot in the absence of snow.

Of the seasonal factors discussed above, only two would be significant in affecting choice of infiltration routes in the Indochina area chosen for application of this model: rains which flood infiltration routes and deteriorate foodstuffs, and winter cold which requires the infiltrator to build fires for warmth and food preparation.

R-656

II. DATA COLLECTION AND INTERPOLATION

In order to apply the model to a real infiltration problem, we chose a rectangular area in Indochina, 210 × 220 km. This area includes part of northern South Vietnam and adjacent Laos lying between 15° and 17° latitude and 106° and 108° longitude (Fig. 1).

The dynamic-programming technique to be used for minimum-cost path determination requires that a finite set of nodal points be designated and the cost from traveling from any point to an adjacent point be specified. The number of nodal points is limited by computer capacity and costs of data collection and computation.

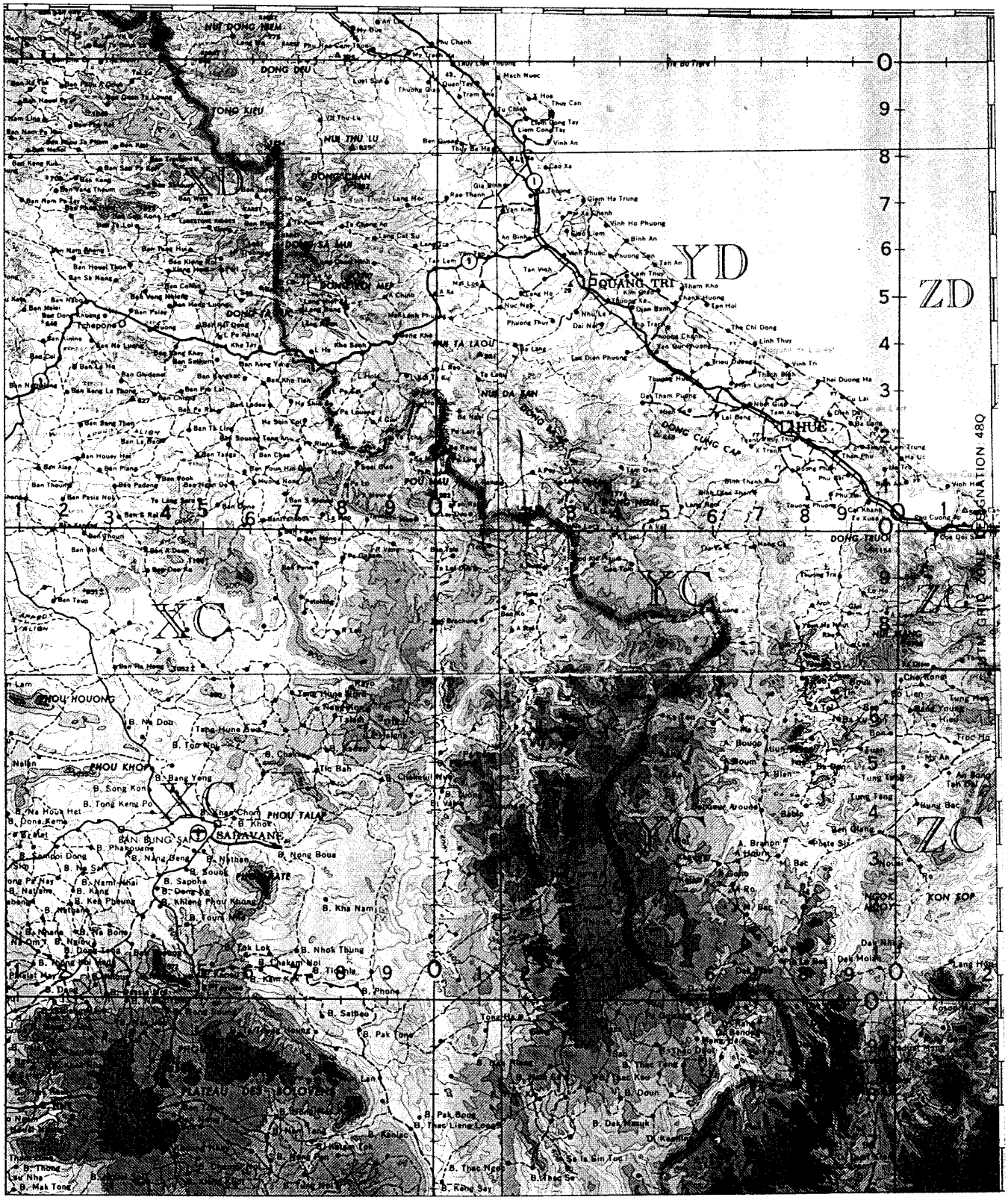
For convenience, we chose a rectangular grid of nodal points. The distance between nearest points was set at 2.5 km because of computer capacity. Because of the time required to collect altitude data over a grid of approximately 8000 points, it was decided to collect data at a smaller number of points and interpolate.

Several techniques were tried for collecting the altitude data in digital form. A graphical input device, the Rand Tablet, was used to trace contour lines from maps of scale 1:250,000, altitudes at grid points being determined by interpolation from altitudes where contour lines crossed grid lines. It was found, however, that this tracing technique was very time-consuming, and the required programming for interpolation was more complex than first appeared.

The simplest way to collect altitude data was the one finally used. A set of plastic relief maps of the area chosen at a scale of 1:250,000 was employed. These maps had altitude contours printed at 100-m intervals. Each 500-m contour line was labeled. The raised relief feature of the map allowed us to determine quickly whether a given contour line was higher or lower in altitude than a labeled line. Also printed on the maps were the standard 10-km grid lines. The points at the intersections of the grid lines were chosen for altitude readings. Altitude at grid points was estimated to the nearest 10 m, by interpolating by eye, using the distance from the grid point to adjacent 100-m contour lines.

Because there were 506 grid points at which altitude data were obtained and more than 8000 points at which altitude data were required,

R-653
19


XC
YC
ZC
YD
ZD
SAVANNAKHET
PHOU KHOUOK
PHOU HOUONG
PHOU TALAK
PLATEAU DES BOLOVENS
QUANG TRI
HUE
DONG HOI
DONG HA
DONG DEU
DONG CHAN
DONG TA MUI
DONG KHE MEK
NUI DONG HEUM
NUI THU LU
NUI TA LAOU
NUI DA LAN
DONG CUNG CAP
DONG TRUOI
KON SOP
Tchepone
Ban Houei Thao
Ban Sa Nang
Ban Ta Ling
Ban Souang Tong Kham
Ban Ta Seo
Ban Taseo
Ban Houey Hai
Ban Piang
Ban Padong
Ban Pasia Noi
Ban Thouay
Ban Voi
Ban Ta Loy
Ban Ta Ngo
Ban Ta Ngo Noi
Ban Ta Ngo Tai
Ban Ta Ngo Tay
Ban Ta Ngo Dong
Ban Ta Ngo Nam
Ban Ta Ngo Bac
Ban Ta Ngo Trung
Ban Ta Ngo Ha
Ban Ta Ngo Thuong
Ban Ta Ngo Cao
Ban Ta Ngo Thap
Ban Ta Ngo Moi
Ban Ta Ngo Cu
Ban Ta Ngo Lang
Ban Ta Ngo Xom
Ban Ta Ngo Ap
Ban Ta Ngo Thon
Ban Ta Ngo Khu
Ban Ta Ngo Phuong
Ban Ta Ngo Xa
Ban Ta Ngo Huyen
Ban Ta Ngo Tinh
Ban Ta Ngo Thanh
Ban Ta Ngo Thi
Ban Ta Ngo Quan
Ban Ta Ngo Dao
Ban Ta Ngo Nguyen
Ban Ta Ngo Le
Ban Ta Ngo Tran
Ban Ta Ngo Ly
Ban Ta Ngo Vo
Ban Ta Ngo Dang
Ban Ta Ngo Bui
Ban Ta Ngo Do
Ban Ta Ngo Ngo
Ban Ta Ngo Duong
Ban Ta Ngo Ho
Ban Ta Ngo Phan
Ban Ta Ngo Vu
Ban Ta Ngo Dinh
Ban Ta Ngo Trinh
Ban Ta Ngo Mai
Ban Ta Ngo Ton
Ban Ta Ngo Thai
Ban Ta Ngo Hoang
Ban Ta Ngo Long
Ban Ta Ngo Phung
Ban Ta Ngo Quy
Ban Ta Ngo Kim
Ban Ta Ngo Thach
Ban Ta Ngo Ngoc
Ban Ta Ngo Bao
Ban Ta Ngo Chau
Ban Ta Ngo Hoa
Ban Ta Ngo Mai
Ban Ta Ngo Lan
Ban Ta Ngo Cuc
Ban Ta Ngo Tuyet
Ban Ta Ngo Van
Ban Ta Ngo Thi
Ban Ta Ngo Nhi
Ban Ta Ngo Tam
Ban Ta Ngo Tu
Ban Ta Ngo Ngu
Ban Ta Ngo Luc
Ban Ta Ngo Thap
Ban Ta Ngo Bach
Ban Ta Ngo Nhat
Ban Ta Ngo Nguyet
Ban Ta Ngo Tinh
Ban Ta Ngo Hanh
Ban Ta Ngo Dung
Ban Ta Ngo Trung
Ban Ta Ngo Hieu
Ban Ta Ngo Nghia
Ban Ta Ngo Le
Ban Ta Ngo Tri
Ban Ta Ngo Tin
Ban Ta Ngo Nhan
Ban Ta Ngo Thanh
Ban Ta Ngo Thuy

several interpolation schemes were investigated. This investigation was conducted by using a JOSS program* which allowed quick implementation of an interpolation technique and provided a map printout on which contour lines could be drawn and compared to those on a geodetic map.

The final interpolation technique chosen because of its resemblance to the real map is analogous to the SYMAP[1] technique developed at Harvard for interpolation of variables over two-dimensional surfaces. Each point to be interpolated is given an altitude which is a weighted average of the altitudes of the 12 adjacent altitudes on the primary 10-km grid. The weighting function is the inverse square of the distance. In other words, the altitude of the point, P, under consideration is proportional to the sum of the altitudes at each adjacent grid point divided by the square of the distance between P and the grid point. The weighted sum is then divided by the sum of the reciprocals of the squares of the distances in order that the weighting factors add up to 1.

This interpolation technique has a certain intuitive appeal. If a cone-shaped mountain were to erode to a cone with a larger base, the altitude at a point halfway from the center of the cone would decrease as the inverse square of the distance of this point from the center.

One disadvantage of this interpolation technique is that no altitude at an interpolated point can be a local maximum (or minimum) point. That is, no mountain peak interior to the grid can be predicted by this technique. Interpolation techniques that extrapolated altitude trends to correct this deficiency were found to yield unrealistic results because the terrain has considerable random structure at scales smaller than the 10-km grid size.

The failure of the technique to predict local minima or basins was not viewed as a handicap because of the relative absence of these features in the terrain being examined. The ability of the technique to structure ridges and valleys accurately is probably very poor, however,

---

*The JOSS program is given in Appendix A. JOSS is the trademark and service mark of The Rand Corporation for its computer program and services using that program. The JOSS system is an on-line, time-shared system developed at Rand.

and this is a major concern for infiltration-route prediction because of their importance. This can be compensated for to a certain extent by manipulation of the cost function, which we will discuss later.

As an example of the ability of the inverse-square interpolation technique to reproduce terrain, we will consider a 40 × 40 km area of South Vietnam containing the A Shau valley. Part (a) of Fig. 2 is a reproduction of the 300-m contour lines drawn from a map of scale 1:1,000,000 and enlarged. The elevation in meters is shown by the accompanying legend. The "T"-shaped region above center between 300 and 600 m elevation is a segment of the A Shau valley. The low area (less than 300 m elevation) in the upper right-hand corner of the figure leads to the coastal plain.

Part (b) of Fig. 2 shows a JOSS printout of the altitudes covering the same area.*

The single digit in every other column represents the altitude rounded to 100 m. For example, the 5 in the lower left-hand corner indicates the altitude at this point is between 500 and 600 m. We drop the first digit for altitudes above 1000 m; for example, 4, the 19th digit in the bottom row, represents 1400 to 1500 m.

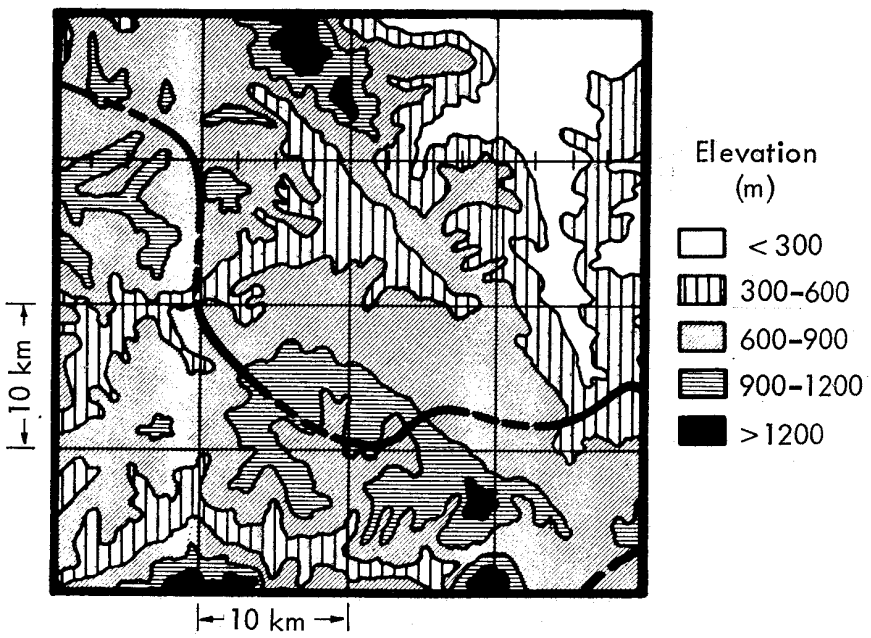
Altitudes were read into the program at the 10-km UTM (Universal Transverse Mercator) grid intersections in Fig. 2(a) as well as at the 24 intersections surrounding the figure. Altitudes were interpolated on a 1-km grid at all other points, using the inverse-square technique.

By comparing (a) and (b) of Fig. 2, we can see that the interpolation program follows the gross terrain characteristics rather well. The high-altitude regions at the bottom of part (a) are reproduced as a result of their falling on grid intersections. The low-altitude region in the upper right-hand corner is well reproduced. The valley areas, however, are not reproduced as well as was expected.

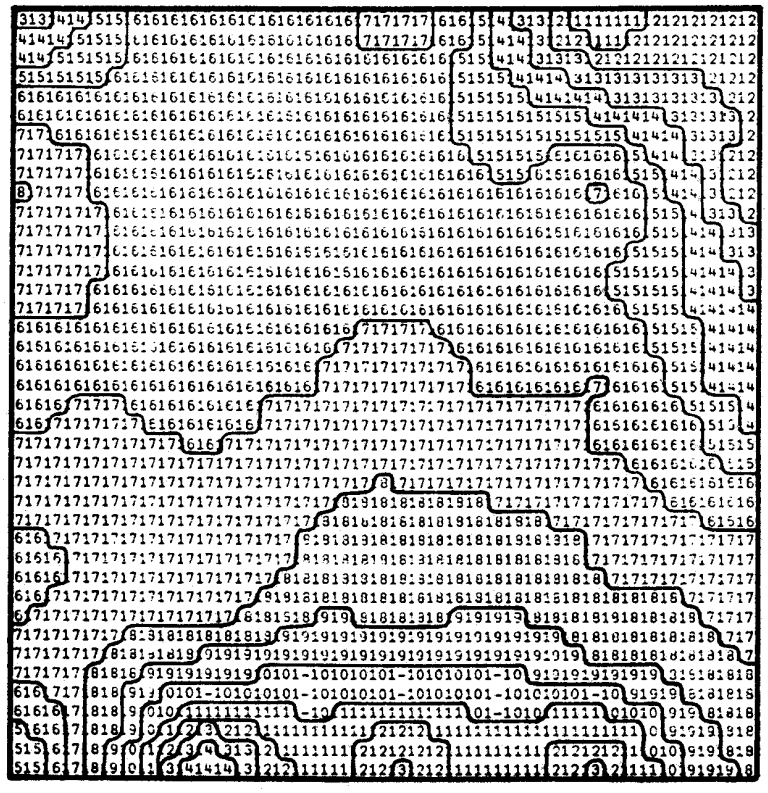
The altitudes on the 10-km grid were obtained from maps of larger scale (1:250,000) than that of Fig. 2(a). It has been observed that these maps do not always agree on the altitude at a grid point, and

---

*Only the digits in the odd-numbered columns are significant. The 1's in the even-numbered columns are inserted only for formatting purposes.



(a) Map contours



(b) Interpolation results

Fig.2—Interpolation

this will account for some of the discrepancy between the interpolated map and the maps used for illustration in this Report. It has also been found that maps of the same scale do not always agree, and this is due in part to uncertainties in altitude of some of the regions of Vietnam when the maps were prepared. Because terrain data used in this study are of limited accuracy, details of the routes predicted by the model are not to be taken very seriously. The data used are considered sufficient to illustrate the value and applications of this approach but are insufficient for strategic or tactical use.

Although our discussion has been concerned primarily with terrain data, other kinds of data can be quantified and interpolated in the same manner. For example, at each grid point an integer can be assigned indicating the amount of vegetative cover for the infiltration route in the area. Heavy jungle can be assigned a value, 1; light cover with occasional trees or brush can be assigned a value, 2; no cover can be assigned a value, 3. And these values can be interpolated on the grid in the same manner as the altitude data. The interpolated values can then be used in the route cost function as described later.

Population data can be handled in a slightly different fashion. The map coordinates and population of each village can be read into the computer program. A number can be assigned to each interpolation point that gives a measure of population density by adding the populations of each village within a given radius of the point (10 km, for example) weighted by the inverse square of the distance from that village to the point. The resulting number can be used as a population cost factor when routes are chosen to avoid (or seek) population. A positive or negative factor could be used for each village, depending on whether it is regarded as unfriendly or friendly. Similarly, one can treat other geographically distributed factors that may be important to choice of infiltration route.

III. THE COST EQUATION

The central hypothesis of our model is that infiltration routes are chosen to minimize some cost, and that this cost accumulates incrementally along the route. In other words, the routes are solutions to minimum-path problems.

In order to solve a minimum-path problem by a dynamic-programming algorithm, it is necessary to specify a cost equation or weighting function, giving the incremental costs along arcs in the grid. We might expect that in the real world such a cost equation would be a complex function of terrain, vegetation characteristics, population distribution, and other factors already discussed. Nevertheless, because we felt terrain was the primary factor determining infiltration routes in mountainous areas, we chose to examine the route implications of a cost equation based on terrain only. This approach does not exclude other cost factors from consideration, since they can be added to the cost equation once the influence of terrain is understood.

In the cost equation, we chose to include three terrain factors:

- The distance between adjacent nodes
- The altitude difference between adjacent nodes
- The roughness of the terrain at each node

We had a choice of different ways to account for the cost of these factors:

- The physical stress on the infiltrator
- The food and water requirements of the infiltrator imposed by the terrain
- The time required to traverse the terrain on foot
- The costs to the logistics system imposed by the terrain

In this case, we chose to use as the cost metric the number of calories required by the infiltrator to traverse the terrain at an efficient rate of speed.* It was felt that the total stress on the individual

*A rate of speed at which the number of calories required per mile of travel is at a minimum.

would be proportional to the work done in traveling. All other things being equal, the food and water and hence logistics requirements would be proportional to the work done. Finally, since there is a maximum rate at which each individual can expend energy on a continuous basis without succumbing to exhaustion, the time required for a given route should be roughly proportional to the total energy expended.

Table 1* shows the rate of expenditure of energy for various traveling conditions. As would be expected, these rates are dependent on the speed of travel. This can be seen more clearly in (a) of Fig. 3,* where a line of 45-deg slope represents a rate of energy expenditure per unit distance. The most efficient walking speed is approximately 3.5 mi/hr; at this speed, energy is expended at a rate of 1/20 kcal per meter of distance traveled.**

These data were obtained for men of various physical conditions traveling on flat, hard surfaces. The effect of traveling uphill can be inferred from treadmill data, shown in (b) of Fig. 3.* It can be seen in this figure that traveling up a 10 percent grade at 3.5 mi/hr requires nearly twice the energy expenditure per mile (or per hour) as traveling on level ground. It is also suggested in the running curves that the energy expenditure (per mile) increases faster than linearly with grade. We have examined both linear and quadratic dependence on grade, as discussed below.

A treadmill is a kind of idealized condition for measuring energy expenditure, since little energy is required to maintain balance or to compensate for uneven terrain. As the data in Table 1 suggest, a greater energy expenditure is required in stubble and plowed fields and in rough country than that shown in Fig. 3.

Most studies neglect to report the energy requirements to descend on a negative grade. There is some evidence, and experience suggests, that a slight downhill grade requires less effort per meter than

---

*Data taken from *Bioastronautics Data Book*.(2)

**Since a pound of rice contains approximately 1600 kcal, we obtain a "mileage" of 32 km per pound of rice. This rate will be smaller for lighter-weight individuals carrying no pack.

Table 1

ENERGY EXPENDITURES OF ACTIVITIES

Activity		Typical values for lb O_2/hr	kcal/min	Btu/hr
Moving over rough terrain on foot				
Flat firm road	2.5 mph	0.11-0.19	2.8-4.9	660-1140
Grass path	2.5	0.12-0.20	3.2-5.1	760-1240
Stubble field	2.5	0.16-0.23	4.0-6.1	960-1440
Deeply plowed field	2.0	0.19-0.27	4.9-6.9	1160-1640
Steep 45° slope	1.5	0.19-0.27	4.9-6.9	1160-1640
Plowed field	3.3	0.30	7.8	1850
Soft snow, with 44-lb load	2.5	0.79	21.0	4850
Load carrying				
Walking on level, with 58-lb load, trained men	2.1 mph	0.07	1.9	450
	2.7	0.11	2.9	690
	3.4	0.18	4.6	1100
	4.1	0.32	8.3	1980
Walking on level, with 67-lb load, trained men	2.1	0.09	2.3	550
	2.7	0.11	2.9	690
	3.4	0.20	5.1	1210
	4.1	0.33	8.4	2000
Walking on level, with 75-lb load, trained men	2.1	0.10	2.5	600
	2.7	0.13	3.4	810
	3.4	0.20	5.2	1240
	4.1	0.34	8.6	2100
Walking up 36% grade, with 43-lb load, sedentary men	0.5	0.26	6.7	1590
	1.0	0.47	12.3	2910
	1.5	0.62	16.0	3800
Movement in snow				
Skiing in loose snow	2.6 mph	0.32	8.1	1930
Sled pulling--low drag, hard snow	2.2	0.34	8.6	2020
Snowshoeing--bearpaw type	2.5	0.34	8.7	2070
Skiing on level	3.0	0.35	9.0	2140
Sled pulling--low drag, medium snow	2.0	0.38	9.7	2310
Snowshoeing--trail type	2.5	0.40	10.3	2460
Walking, 12-18" snow, breakable crust	2.5	0.50	12.7	3010
Skiing on loose snow	5.2	0.52	14.6	3800
Snowshoeing--trail type	3.5	0.59	14.8	4200
Skiing on loose snow	8.1	0.80	20.6	4900
Measured work at different altitudes				
Bicycle ergometer 430 kg-m/min	720 mm Hg	0.20	5.1	1230
430	620	0.19	4.9	1170
430	520	0.21	5.4	1290
Mountain climbing 880-1037 kg-m/min	610 mm Hg	0.36-0.43	9.2-11.0	2200-2640
566- 786	425	0.30-0.37	7.7- 9.5	1840-2260
393- 580	370	0.25-0.41	6.4-10.5	1530-2520

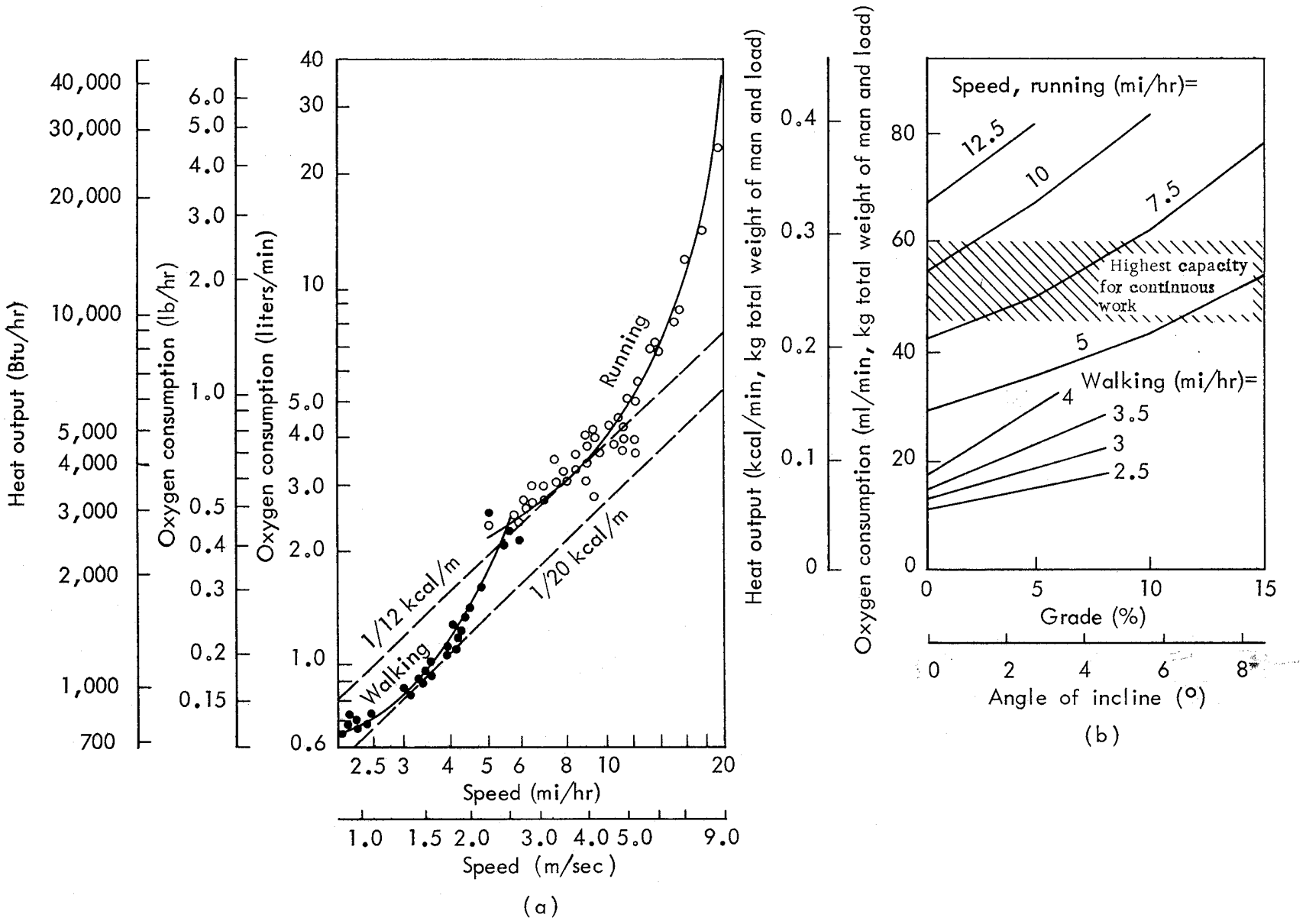


Fig.3—Energy costs of foot travel

horizontal travel. The situation changes, however, when the downhill grade increases to the point where slippage is possible. In this case considerable energy can be required to maintain footing.

Although we experimented with cost functions that included both a linear and a quadratic dependence on grade in order to reflect the lower energy requirement for small negative grades, the results were not entirely satisfactory because of uncertainty as to the relative weights of the linear and quadratic terms.* We resolved this ambiguity by arguing that infiltration routes will be traveled in both directions as personnel are rotated and because food is often transported to camps in the highlands from either direction. With this argument we primarily examined cost functions which depend only on the absolute value of the altitude change between grid points and which neglect the difference between uphill and downhill.

Some mountaineering experts have measured the energy required for traveling various kinds of terrain in many kinds of weather.(3) Their results were obtained by use of an oxygen-consumption meter strapped to the back during their many trips. Their results were fitted to an empirical equation to provide a means of estimating food and rest requirements in advance of any given trip. The equation agrees very well with the *Data Book* results given above:

$$E = 100(10 + R + 2C + 4H)$$

where

E = energy expenditure (kcal/day)
R = distance on roads (mi/day)
C = distance cross-country (mi/day)
H = height climbed (kft/day)

Energy expenditure for road travel is 1/20 kcal/m, and energy-rate expenditure for a 10 percent grade is twice that on a horizontal path cross-country.

Reference 3 indicates that 4000 kcal/day is about the maximum rate

*And for other reasons discussed in the next section.

of expenditure to be expected on a continuous basis by a young, 140-lb man in good condition. If the man is heavier than 140 lb or is carrying a backpack, the value of E must be increased proportionately.

One of the first cost functions used in this model was of the form of Eq. (1):

$$\text{Cost} = C[1 + a(\text{grade}) + b(\text{grade})^2] \tag{1}$$

where C = 250 kcal, the energy required to travel between adjacent grid points (2.5 km) on flat terrain (1/10 kcal/m). By choosing a + 10b = 10, we can double the energy required for 10 percent upgrades, and by making b large or small we can adjust the cost of steeper grades.

This form of the cost function is not as useful as it appears. Since the cost for the total route is the sum of the costs between grid points and grades are proportional to altitude changes, between the same start and end points the linear term contributes the same cost to all routes, because the total cost due to this term depends only on the altitude difference between the start and end points. Similarly, the constant C will not change the choice of routes.

A more useful form of the cost equation is

$$\text{Cost} = 1 + (\alpha|\text{grade}|)^p \tag{2}$$

By taking the absolute value of the grade we do not obtain the same result for different routes when the power p is 1. Values of 1 and 2 were investigated for p; and values of 10, 20, and 40 for α. The discussion of energy requirements suggests a value for α = 10 to double the rate of energy consumption on a 10 percent grade. This value of α is too low because of the interpolation process.

Figure 4 is a graph of the mean-square difference in altitude at two grid points separated by a distance ℓ. Two curves are calculated, one using interpolated altitudes and another using altitudes read directly from the map at 2.5-km intervals. Since our interpolation is a smoothing process, we would expect interpolated altitude differences to be less than the actual differences. At the 2.5-km grid distance, the

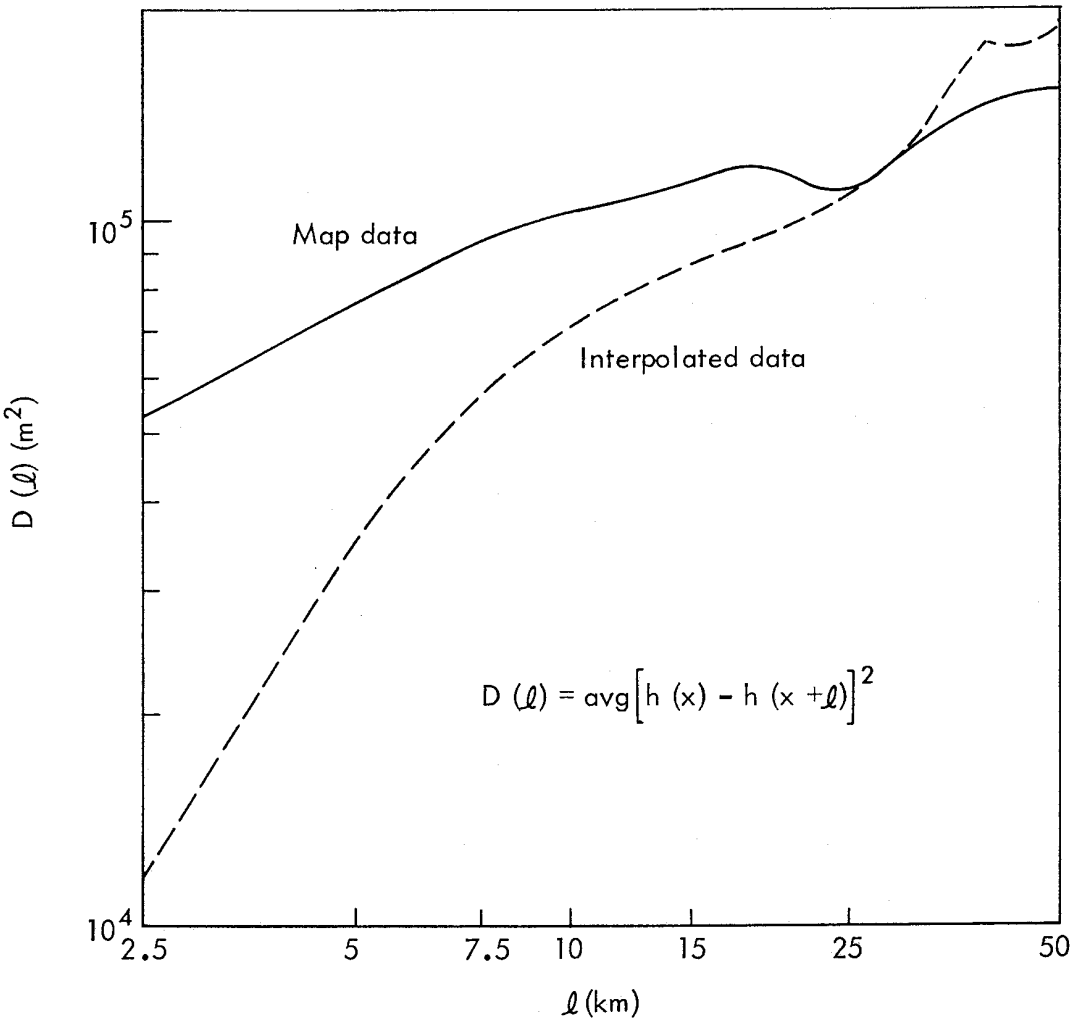


Fig.4—Altitude structure function

mean-square difference is a factor of 5 less than the map data. This can be accounted for by a factor of 2 or more in α.

Furthermore, since the real terrain between grid points is not a smooth ramp, extrapolation of the map data structure function to smaller intervals indicates that the true root-mean-square altitude difference does not drop as fast as the interval distance, and hence the average grade (absolute value) is larger still. Using a 1-km grid distance would have required increasing α by another factor of 2. We feel, therefore, that $\alpha = 40$ is a more realistic choice if the cost function is to be proportional to energy consumption.

Because we found so much evidence in the literature and in conversation with people acquainted with infiltration tactics that ridge routes and valley routes are to be preferred to travel along hillsides or mountainsides, we chose to add a term to the cost equation which penalized grid points that had steep gradients, independent of the relation of the gradient direction to the direction of travel. The term added we call the roughness factor, R, where

$$R = (\beta \text{ grade})^2$$

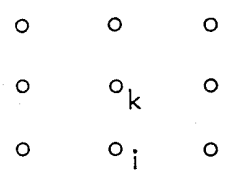
and the square of the grade at a grid point P is found by adding the square of the altitude difference of the two grid points north and south of P to the square of the altitude difference at the two points east and west of P and dividing by 4 times the square of the grid distance. Values of β of 20 and 40 were used in order that the calculated grade using interpolated data would compare to the real grade. This roughness factor compensates in part for the fact that the interpolation technique underestimates valleys and ridges, as was discussed in an earlier section. Its use in the cost equation is purely ad hoc, however, since few data have been found which relate energy saved using ridge and valley routes rather than hillsides except for the fact that hillsides will often be rougher ground due to erosion. There are important exceptions, however. Ridgelines on new mountain ranges can be extremely rugged. Nevertheless, this does not appear to be the case in Southeast Asia, where the mountains are older and more eroded.

IV. THE COMPUTER MODEL

PREMIR is primarily a mathematical computer model which in its present form takes into account terrain data from a geographical area and by means of a suitable weighting function arrives at a set of predicted (optimal) infiltration routes between various start and specific end points. A total cost measure is assigned to each route according to its computed degree of difficulty. This section will describe the details of the computer model and how optimal routes are determined.

A basic underlying assumption in the development of the model is that topography, as well as other terrain factors, is an important determinant in the choice of routes. Thus, input to the model includes altitude readings from a chosen geographical area arrayed in a rectangular grid format. A two-dimensional interpolation function is employed to interpolate for points between the input grid data points, resulting in a finer grid representation of the area under study.

The above procedure facilitates the construction of a somewhat comprehensive network representation of the study area consisting of nodes and arcs. The nodes (cross-points) of the network are the input and interpolated data points. Each interior point in the network has eight surrounding points, and each exterior point has either three or five surrounding points, depending on its position. An example of a grid network is shown below; i is an exterior point, and k, an interior point.



Movement within the network is permitted between each point and its surrounding points via arcs. Arcs are characterized by their beginning and ending nodes and an associated cost.

A weighting function is used to compute the cost of moving from one point within the network system to another as a function of the

change in altitude. Also available is an option to include in the weighting function a measure for the roughness of the area within which a data point lies. All of the data-point values (input and interpolated) and cost calculations may be printed as output from the model.

A dynamic-programming minimum-path algorithm is then applied to find minimum-cost (optimal) paths between all possible start and specific end points of the system. The algorithm encompasses a value-iterative scheme, and the solution which is produced gives the best node to go to from each nodal point in the system en route to a specific terminal point. Also obtained is a cost measure at each nodal point representing the total cost of an optimal route from that point to the terminal.

The model is written in FORTRAN IV and under its present configuration allows for multiple cases to be included during a single computer run, subject to the time constraints imposed by the operating system. Computer storage requirements are mainly dependent upon the total number of grid data points (both input and interpolated) for the study area. Currently, a case resulting in a 100 × 100 grid array* requires 444k bytes of storage (k = 1024) on the IBM 360 system. Smaller-sized problems require less storage space. Copies of the computer source deck or object deck (compiled on the optimal compiler) may be obtained from the authors.

PREMIR is structured into modules with each one performing different tasks. The following is a functional description of the principal modules (subroutines) and underlying rationale.

MAIN PROGRAM

The main program functions as executor for the model. Control information relating to the problem case as well as the input data on the study area is read into the model by this routine. It also calls on other routines to perform certain tasks as specified by the user. The main program provides for special efforts to detect input errors and identify them for the user.

*Such an array contains 10,000 data points.

INTERPOLATION

The user may choose to use only the input data or have the model interpolate for a variable number of additional data points in creating the network representation of the study area. In the latter case, a two-dimensional interpolation algorithm is employed. The algorithm interpolates purely on the basis of an inverse-squared distance weighting function. A function of this type not only gives seemingly satisfactory empirical results, but also presents an easy calculation. In this function, the value at any point, say P, is a weighted average of the values of the input data points within a neighborhood of P.

Here we let Z_i be the value at input data point D_i, and R_i be the cartesian distance between P and D. The value at point P, considering N nearby input data points somewhat symmetrically spaced about P, is given by the following:

$$f(P) = \begin{cases} \sum_{i}^{N} (R_i)^{-2} Z_i \Big/ \sum_{i}^{N} (R_i)^{-2} & \text{for } R_i \neq 0 \text{ and all } D_i \text{ within the neighborhood} \\ Z_i & \text{for } R_i = 0 \text{ and some } D_i \text{ within the neighborhood} \end{cases} \qquad (3)$$

COST CALCULATION

In order to obtain a set of predicted (optimal) routes, a method of categorizing all possible paths according to their relative degree of difficulty must be used. We achieve this categorization by assigning a cost to travel between the different nodes (data points) in the network system representing the study area. Thus, each allowable arc in the system has an associated cost, and the total cost assigned to a route from various points in the network to a specific end point is merely the sum of the costs for the arcs included in the route.

The structure of PREMIR facilitates the testing of various cost functions which take into account topography and other relevant data. In the present application, several different cost function formats and parameters were investigated. The model as presently programmed employs a cost function based on the change in altitude between nodal

points and an option is available to include a weighting for the roughness of the area within which a data point lies.

The cost for traveling from one node, say node i, within the network system to some adjacent node j is given by the following:

$$\text{Cost } (i, j) = \begin{cases} 1 + c\left(\alpha \dfrac{\Delta \text{ altitude}}{G}\right)^p & \text{for two nodes which are nondiagonally separated} \\ 1.4 + c\alpha\left(\dfrac{\Delta \text{ altitude}}{G}\right)^p & \text{for two nodes which are diagonally separated} \end{cases} \tag{4}$$

where c, α, and p are different parameters of the cost function; G is the nondiagonal (horizontal or vertical) distance in meters between nodes of the rectangular network.

If the option to include the roughness factor into the cost calculation is elected, the following procedure is used. Let (x, y) be the cartesian coordinates of node j. Then to Eq. (4) above we add the following quantity:

$$[\beta(\text{altitude } (x + 1, y) - \text{altitude } (x - 1, y))/(2G)]^2 + [\beta(\text{altitude } (x, y + 1) - \text{altitude } (x, y - 1))/(2G)]^2 \tag{5}$$

where β is a parameter of this segment of the cost function.

To insure that predicted routes do not extend beyond the grid array or go into undesired areas (e.g., into the ocean) a blocking technique is used. That is, a special altitude value is assigned to the outside borders of the grid by another subroutine of the model. If a section of the rectangular grid includes a large body of water or other undesired area of travel, the altitude values (data points) in the section(s) can also be flagged. In computing the costs, a check is made, and if the special altitudes are sensed, a very high cost (10^6) is assigned for traveling to that point. An input to the model is an upper bound for the special altitudes, and all points with altitudes less than or equal to this bound are assigned the high cost.

What effect blockage of certain passageways will have on a set of

predicted routes may be investigated by inputting to the model coordinates of the place(s) to be blocked and assigning a high cost accordingly. However, the cost calculations for neighboring points are also affected when blocking is employed.

DYNAMIC PROGRAMMING

A network structure consisting of, say, p nodes with directed arcs connecting some of the nodes and costs assigned for traversing the various arcs is amenable to solution by the techniques of dynamic programming. The term *dynamic programming* is frequently applied to the mathematical analysis of problems in which conditions that must be satisfied by an optimal time-staged decision process are to be exploited to determine best action.

PREMIR incorporates a dynamic-programming minimum-cost-path algorithm to find optimal routes through the network. The solution--an optimal solution--is defined to be a path (from nodal points in the network to a specific terminal point) that satisfies the following principle of optimality:

> *Regardless of the previous rationale used to arrive at a particular state, the remaining decisions as to what path to take en route to the terminal must themselves constitute an optimal solution.*

Here we let Y_i be the present value (sum of the costs) of an optimal path from node i to the terminal node r, where *optimal* implies a path having minimal total cost. Now, if an optimal path from node i to node r starts by first going to node j, then

$$Y_i = Y_j + C_{ij} \tag{6}$$

and

$$Y_i \leq Y_k + C_{ik} \tag{7}$$

for all $k \neq j$ in the system.

Equation (6) states that for an optimal path from node i to the terminal, r, which starts by going to node j, the cost is the sum of the cost of an optimal path from node j to r and the cost of going from node i to node j. Equation (7) insures that among the possible choices of paths to take from node i, none is better than the one starting with node j.

Since Eqs. (6) and (7) must hold for every node i different from the terminal node r, then Y_i must satisfy a set of functional equations,

$$Y_i = \underset{\substack{(i,\,j)\\ \text{in the system}}}{\text{minimum}} [Y_j + C_{ij}] \tag{8}$$

for all $i \neq r$, and

$$Y_r \equiv 0 \tag{9}$$

A method of successive approximation may be used to solve for the $p - 1$ unknowns, $Y_i (i \neq r)$. Specifically, a value-iterative algorithm is applied. The value of the Y_j's is initially set at zero, and the Y_i's are then calculated as in Eq. (8) for every node $i \neq r$ in the system. That is, the first iteration is computed as follows:

$$Y(1, i) = \underset{\substack{(i,\,j)\\ \text{in the system}}}{\text{minimum}} [0 + C_{ij}] \tag{10}$$

for all $i \neq r$ with $Y(1, r) \equiv 0$. The first index on Y is the iteration number.

On the n^{th} iteration, the Y_i's computed during iteration $n - 1$ become estimates of the Y_j's, and new Y_i's are computed as follows:

$$Y(n, i) = \underset{\substack{(i,\,j)\\ \text{in the system}}}{\text{minimum}} [Y(n - 1, j) + C_{ij}] \tag{11}$$

again for all $i \neq r$ and $Y(n, r) \equiv 0$.

The quantity Y(n, i) is interpreted as the minimum cost of a path starting from node i that contains exactly n arcs, unless the terminal is on the path, in which case the path terminates at node r. If the cost, C_{ij}, around every loop in the system is positive, then an n^* exists such that for all $n \geq n^*$, $Y(n, i) = Y(n^*, i)$. We terminate the iterative process when this condition is met. That is, after each iteration greater than some q(q > 1), we test Y(n, i) = Y(n - 1, i), and when this condition is met for all i's in the system, we have obtained the optimal solution. We also obtain another vector, g(n, i), which has its value set equal to the j that minimizes Eq. (10) or (11), depending on the iteration number.

That the process will terminate in a finite number of steps is shown by the following argument: As n increases, eventually every path of n arcs reaches the terminal. And when this occurs, Y(n, i) will stabilize to the correct minimal-cost value for the path, since the only new value added to Y(n, i) will be $Y(n, r) \equiv 0$.

There are several algorithms for finding optimal paths[†] through a network. Corresponding to almost any shortest-path algorithm some class of network structures exists for which the algorithm is relatively inefficient. Since all iterative procedures are in some sense inefficient, economic considerations alone demand that care be taken in the selection and adaptation of a shortest-path algorithm.

Dreyfus[(4)] has examined various shortest-path algorithms by developing upper bounds on the maximum number of computational steps--the number of additions and comparisons directly attributable to the algorithm itself--involved in solving a standardized network problem. He considers one procedure to be significantly superior to another when the bounds differ by a multiplicative factor involving p, the number of nodes. By this criterion the algorithm applied in the present study is less efficient than some other applicable algorithms. That is, the algorithm described above has a higher bound on the maximum number of computational steps than some other algorithm for solving the standardized problem. However, this assertion and its implications can be

---

[†]Such algorithms are referred to as shortest-path algorithms where the length of each (i, j) is taken to be the cost C_{ij}.

misleading, since the actual number of computational steps required to find solutions in this investigation falls far short of the bound computed by Dreyfus and the actual number of steps required by some other algorithms. (This result is in part due to certain specifics of the problem and modifications of the algorithm for programming purposes.)

From our limited experience in this investigation it appears that the actual number of computational steps required is more dependent upon the interrelation among the costs, C_{ij}, than upon the upper bounds. Also dynamic-programming algorithms can require a great deal of computer storage and time to solve various problems. Sometimes the size requirements of a particular problem may be such that the modifications necessary to adapt a specific algorithm may themselves render the algorithm inefficient. The inability to specify the actual number of computational steps required is a shortcoming of many shortest-path algorithms.

In real applications of PREMIR, many initial and terminal points will be considered and it may well be inefficient to make a separate run for each case. Also, in a more rigorous study, different variations of the problem posed here may be investigated, such as determining the K^{th} best route or finding the best route visiting specified nodes. Dreyfus also describes algorithms developed by Floyd and Dantzig for finding simultaneously the shortest paths between all pairs of nodes of a network, algorithms for finding the second- (and third-, etc.) shortest path through a network, and a method for finding shortest paths visiting specified nodes. Given the necessary time and resources, worthwhile experience with the model could be gained by exploration with some of these ideas as options or main features.

V. THE VIETNAM EXAMPLE

Since we are limited by computer size to approximately 10,000 grid points, we can choose to examine a large area with a coarse grid or a smaller area with a fine grid. The proper order would seem to be to find the optimal routes over the larger area and examine details of the routes at a later time with a finer grid.

The area chosen is illustrated in Fig. 5. The coordinates closest to the map are the UTM coordinates of the 10-km grid network. The second row of coordinates (at the very bottom and far left) are computer coordinates established for this problem. One unit in these coordinates represents a distance of 2.5 km, the grid size used to establish routes. We shall use these coordinates in the discussion which follows.

The computer run requires that a terminal point be established; minimum-cost paths from every other point on the grid are then computed. Since the optimum path *from* any point is unique and independent of the path *to* that point, it is sufficient for the computer to print out a table which specifies to which of the eight adjacent points the user proceeds from any point on the grid. From this table optimal paths to the terminal point can be traced from any initial point.

In Fig. 5 the terminal point chosen was at coordinates (60, 66), the east/west coordinate being given first. Initial points were chosen every 10 km along the left-hand edge of the grid. The cost function of Eq. (1) was used with a = 10 and b = 50. Since this set of values underestimates the costs of rough terrain, we see very little route selection to avoid rough terrain.

In Fig. 6 we have increased the cost parameters to a = 20 and b = 200. We can see now some further terrain avoidance, but the routes still tend to be rather straight.

Since, as discussed before, the linear term of cost Eq. (1) provides no discrimination on route choice, we performed the remainder of our calculations with cost Eq. (2). Figure 7 illustrates such a case

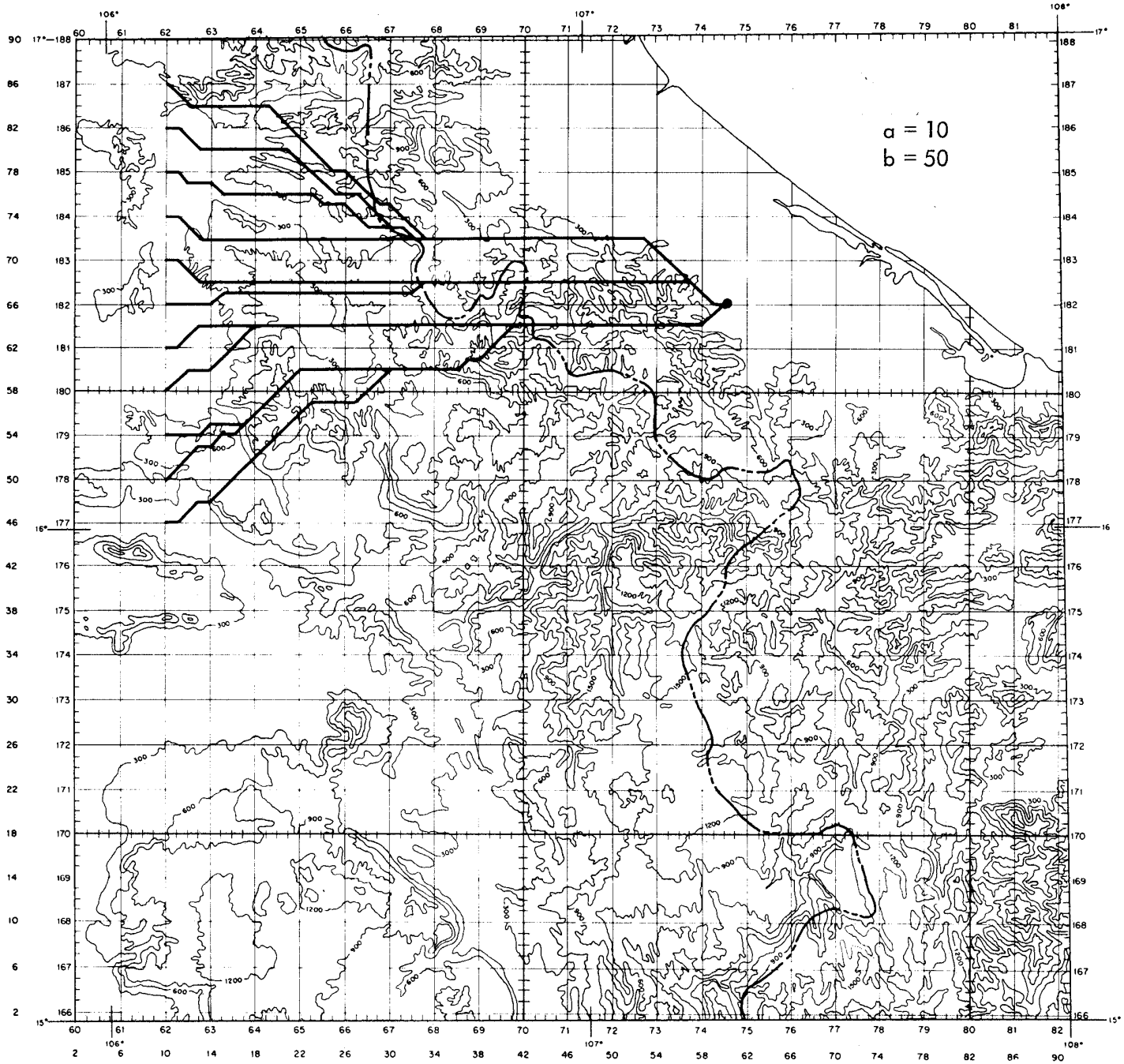


Fig.5—Predicted infiltration routes, with terminal point at coordinates (60, 66)
(a = 10, b = 50)

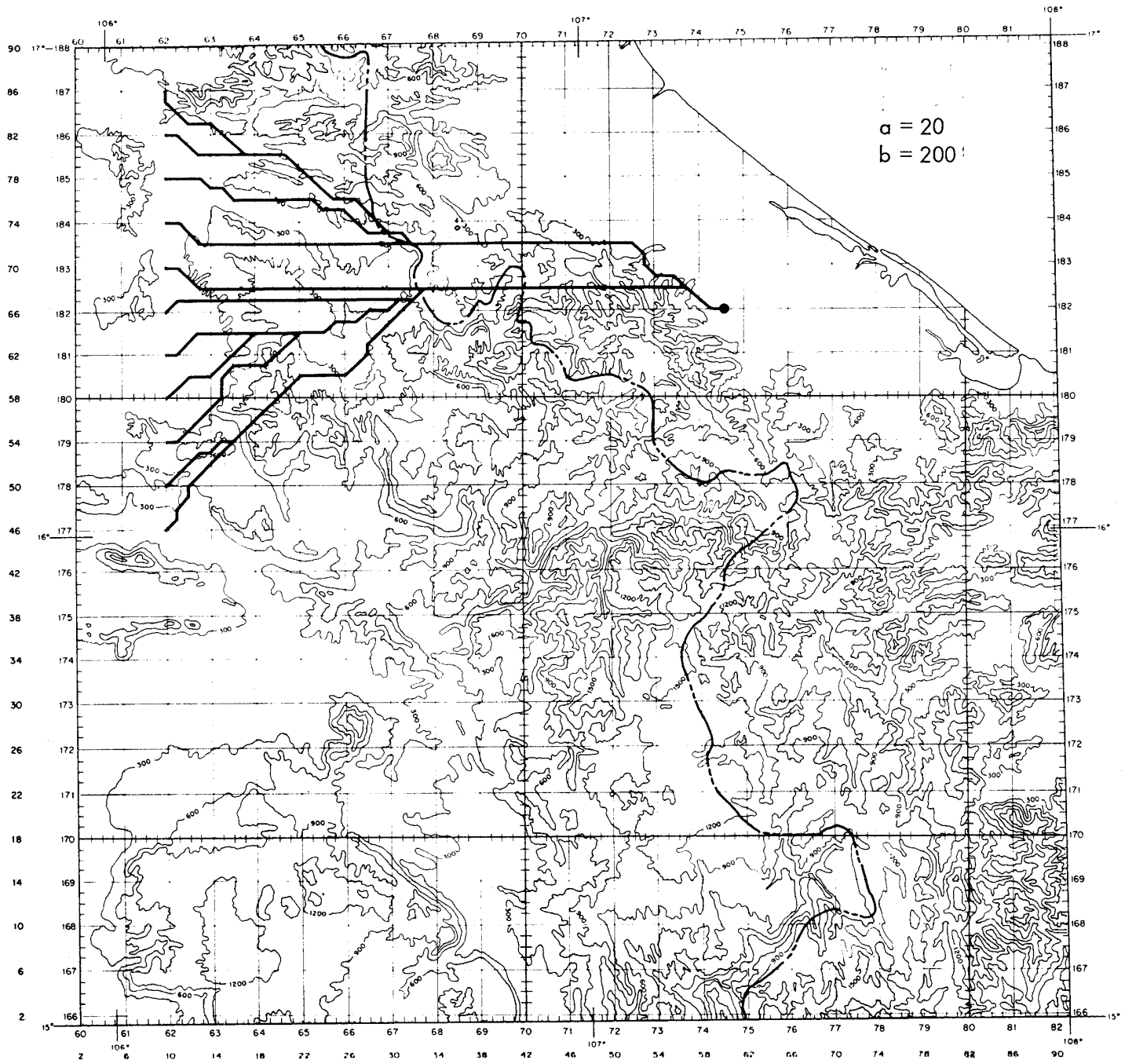


Fig.6—Predicted infiltration routes, with terminal point at coordinates (60, 66)
(a = 20, b = 200)

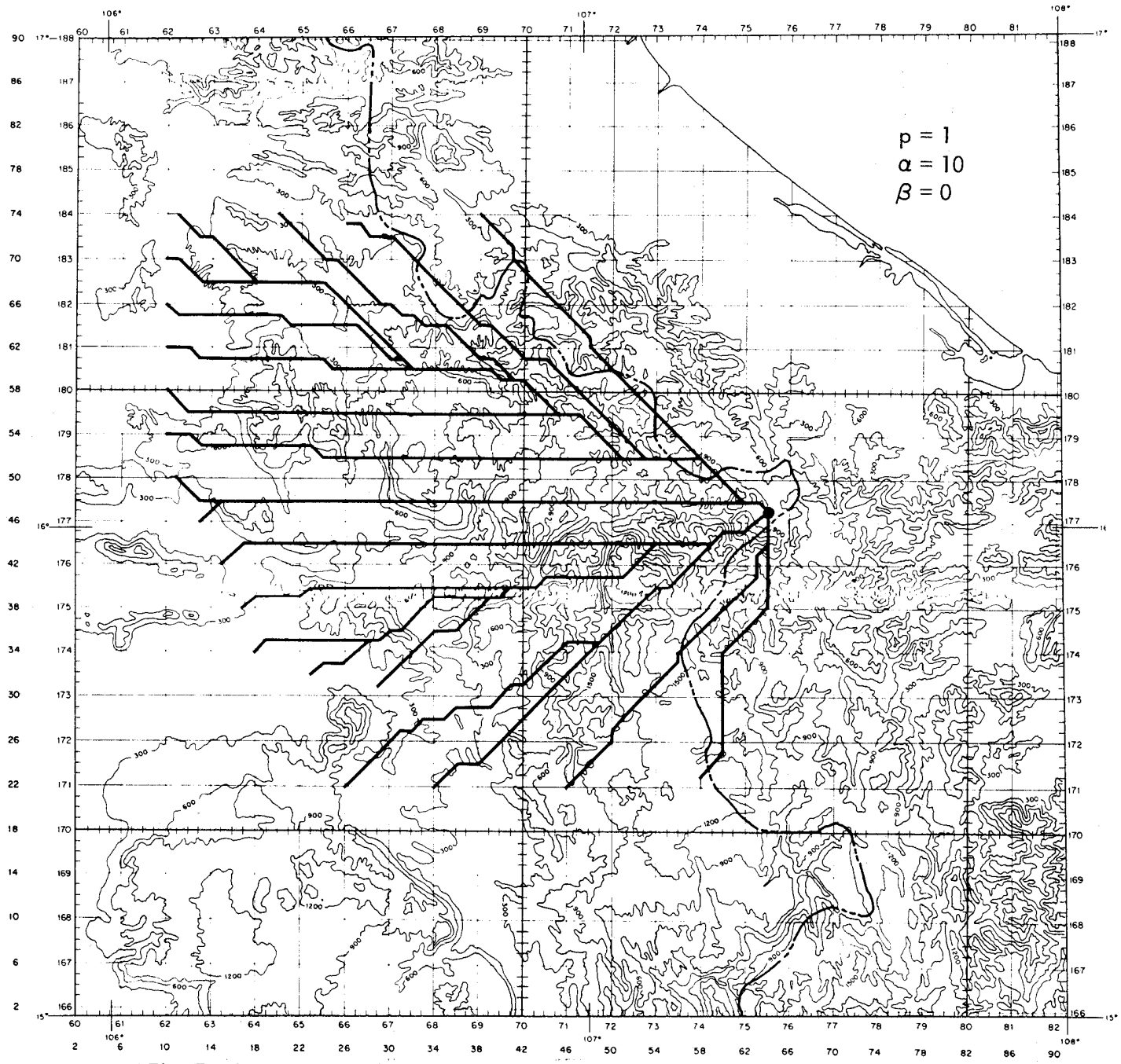


Fig.7—Predicted infiltration routes, with terminal point at coordinates (64, 47)
(p = 1, α = 10, β = 0)

where the parameters have their smallest value, $p = 1$ and $\alpha = 10$. The terminal point is at (64, 47), a likely base area, and the initial points were chosen to lie near to the road network in Laos. The reason for these initial points is that the infiltrators can in theory be brought to these points by truck, and the costs of reaching these points would be on a different basis than the costs for reaching them on foot.

In Fig. 7 there is little terrain discrimination in the route selection because the value of α underestimates the cost of rough terrain. In Fig. 8, however, we have increased α by a factor of 4 to its realistic value and we have added the roughness term to the cost equation with a value $\beta = 40$. Furthermore, grades larger than 10 percent are more heavily discriminated against by setting $p = 2$ so that terrain costs increase as the square of the grade.

In this figure we see considerable selectivity in routes. The northern routes reach into Vietnam to pass along the A Shau valley.* The central route proceeds in a way that is quite surprising in view of the contour lines shown on the map. Especially strange is its crossing the valley at coordinates (44, 44). An examination of the terrain data and their interpolation indicates that our input grid was too coarse for this narrow valley and it does not appear in the interpolation.

In Figs. 9 and 10 we examine another possible terminal point at (70, 18) just inside the Laos border. In these examples we can see the changes produced by adding the roughness factor ($\beta = 40$) to the cost equation for Fig. 10. In both figures, $\alpha = 40$ and $p = 2$. The effect of the factor is to shift many of the central routes to that path which passes along the Vietnam border. The remaining central path shifts to cross the rough terrain at a narrower point. The southerly routes also pick their way more carefully through the rough terrain. Although there is flat terrain available inside Vietnam, the distances involved are such that the routes remain for the most part inside Laos in spite of the mountainous areas.

---

*Southwest of grid coordinates (58, 58).

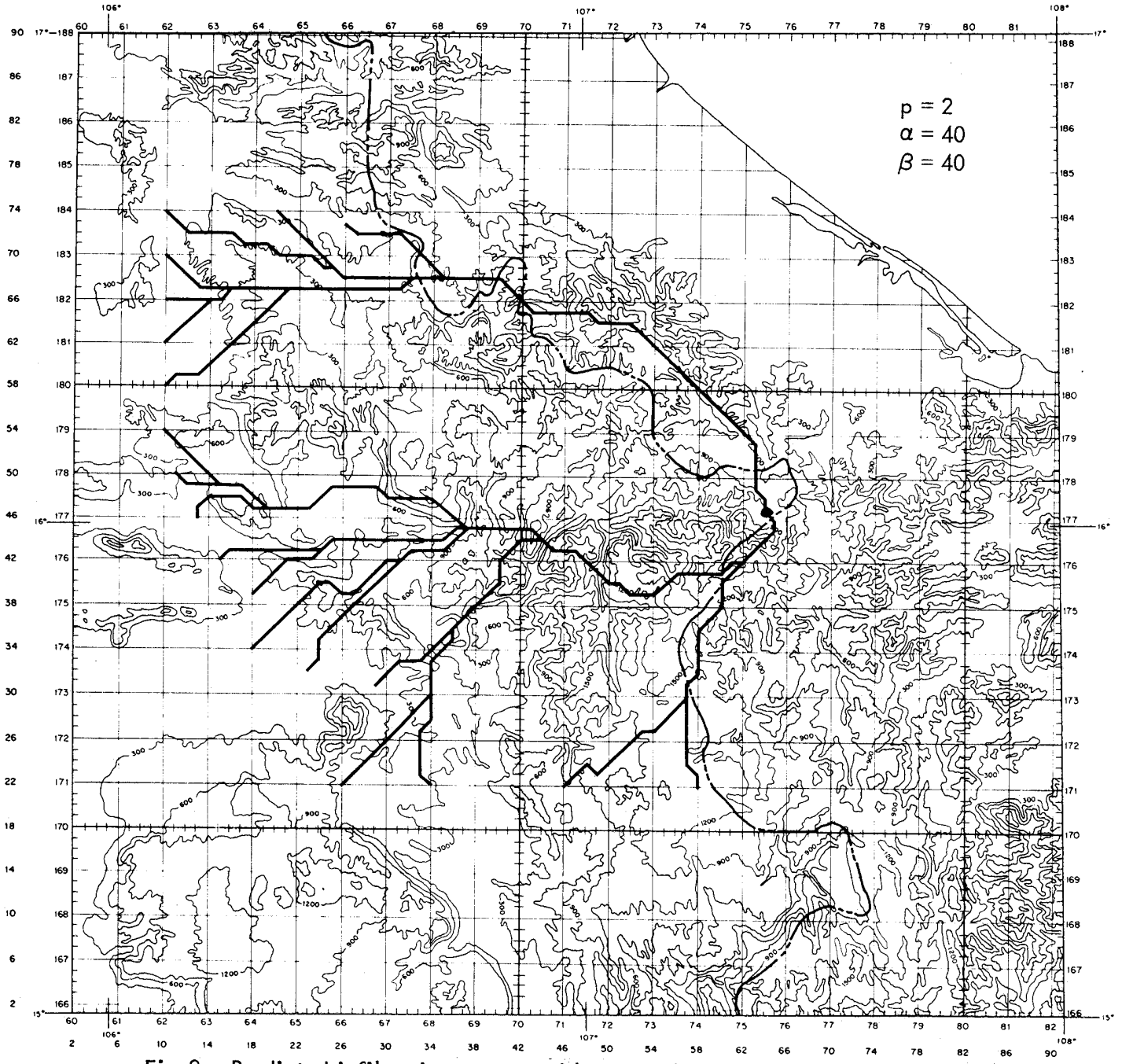


Fig.8—Predicted infiltration routes, with terminal point at coordinates (64, 47)
(p = 2, α = 40, β = 40)

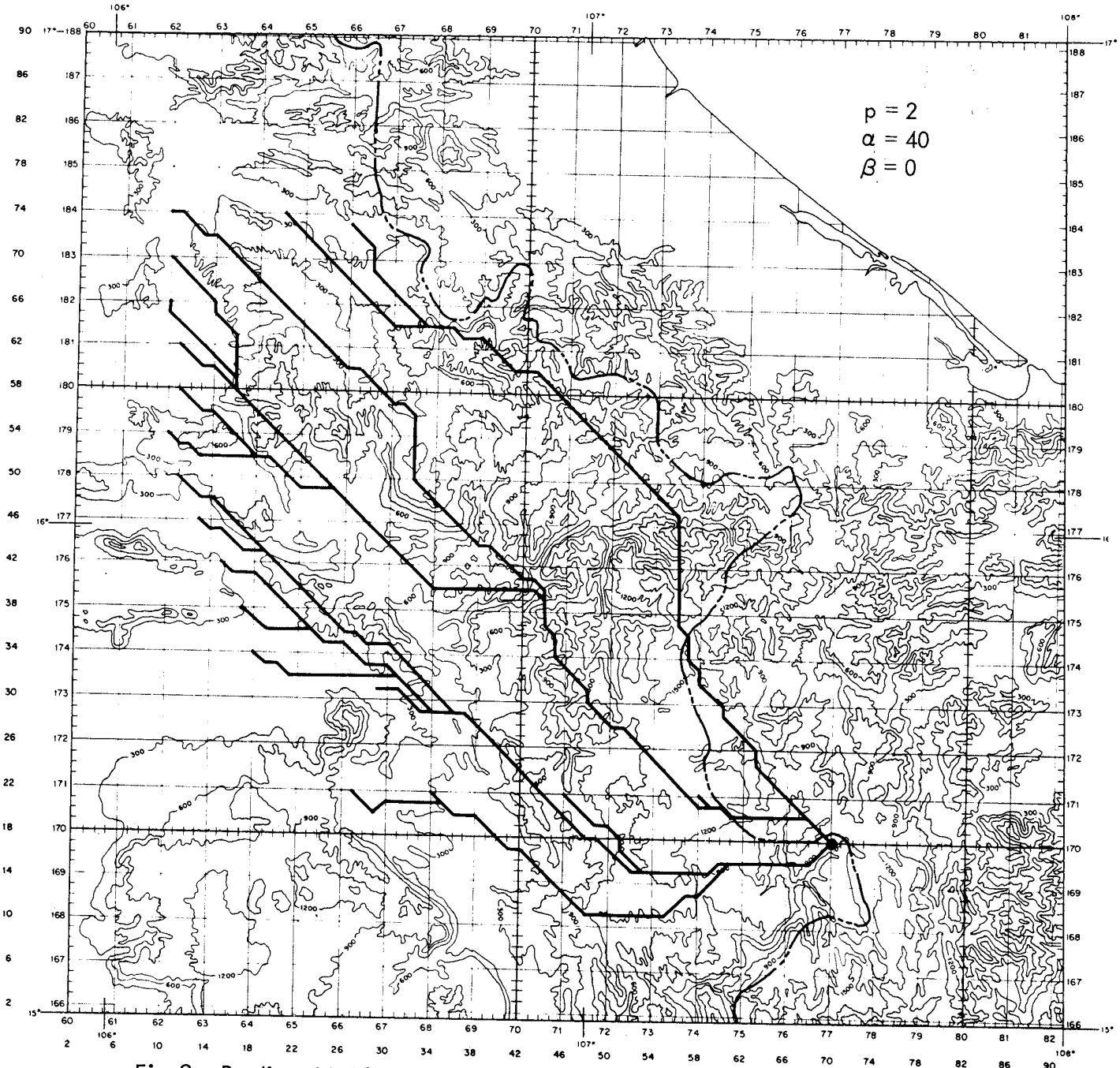


Fig.9—Predicted infiltration routes, with terminal point at coordinates (70, 18)
(p = 2, α = 40, β = 0)

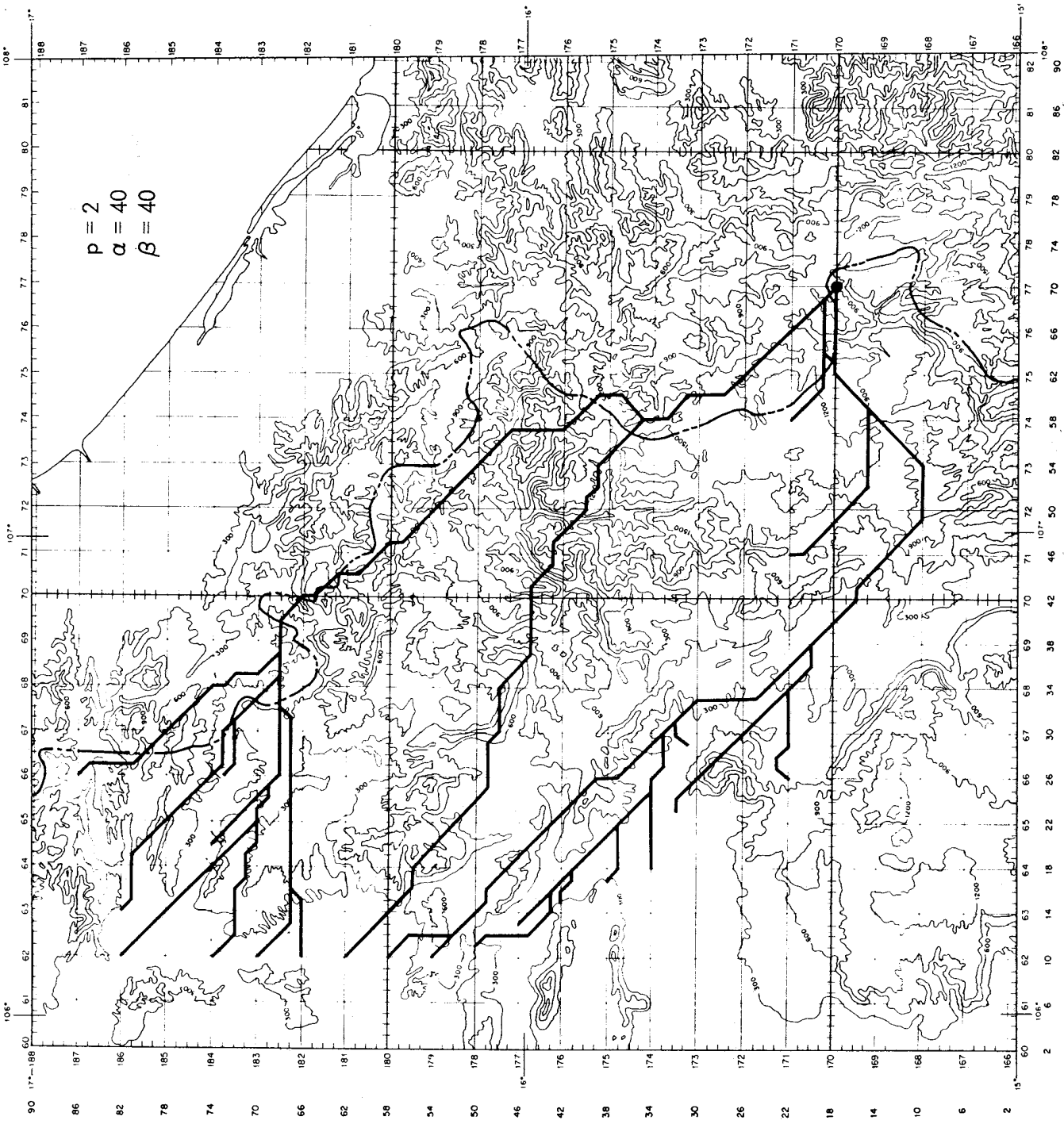


Fig. 10—Predicted infiltration routes, with terminal point at coordinates (70, 18) (p = 2, α = 40, β = 40)

K-656

In Figs. 11 and 12 we examine routes to Hué (72, 67) with and without the roughness factor. On the basis of altitude costs alone there is considerable choice of alternate routes through the mountains, as shown in Fig. 11. Adding the roughness cost in Fig. 12 narrows the choice considerably. In both cases the central routes tend to pass through the A Shau valley in the region of (56, 56). Many of the routes in Fig. 12 move considerably to the north to avoid the mountainous regions in the center.

This effect is also present in Fig. 13 where the terminal point has been moved further west to (60, 66). The general route structure is unchanged.

Because of the isolation of the central route in Fig. 13, it was decided to investigate the effect of a barrier built across this route. To build a barrier in the model it is sufficient to raise the costs at a continuous set of grid points. This was done in Fig. 14 along a diagonal line shown passing through (46, 46). The effect is to cause the central route to move considerably to the north of the original route rather than simply move around the edge of the barrier. In this way the effectiveness of various barrier locations and the minimum size of a barrier necessary to require a major route shift can be investigated.

In Fig. 15 we see the same case but with cross-shaped barriers placed in the Khe Sanh and Quang Tri areas. Because of the flatness of the terrain in these areas, the infiltration routes bypass these barriers with little change.

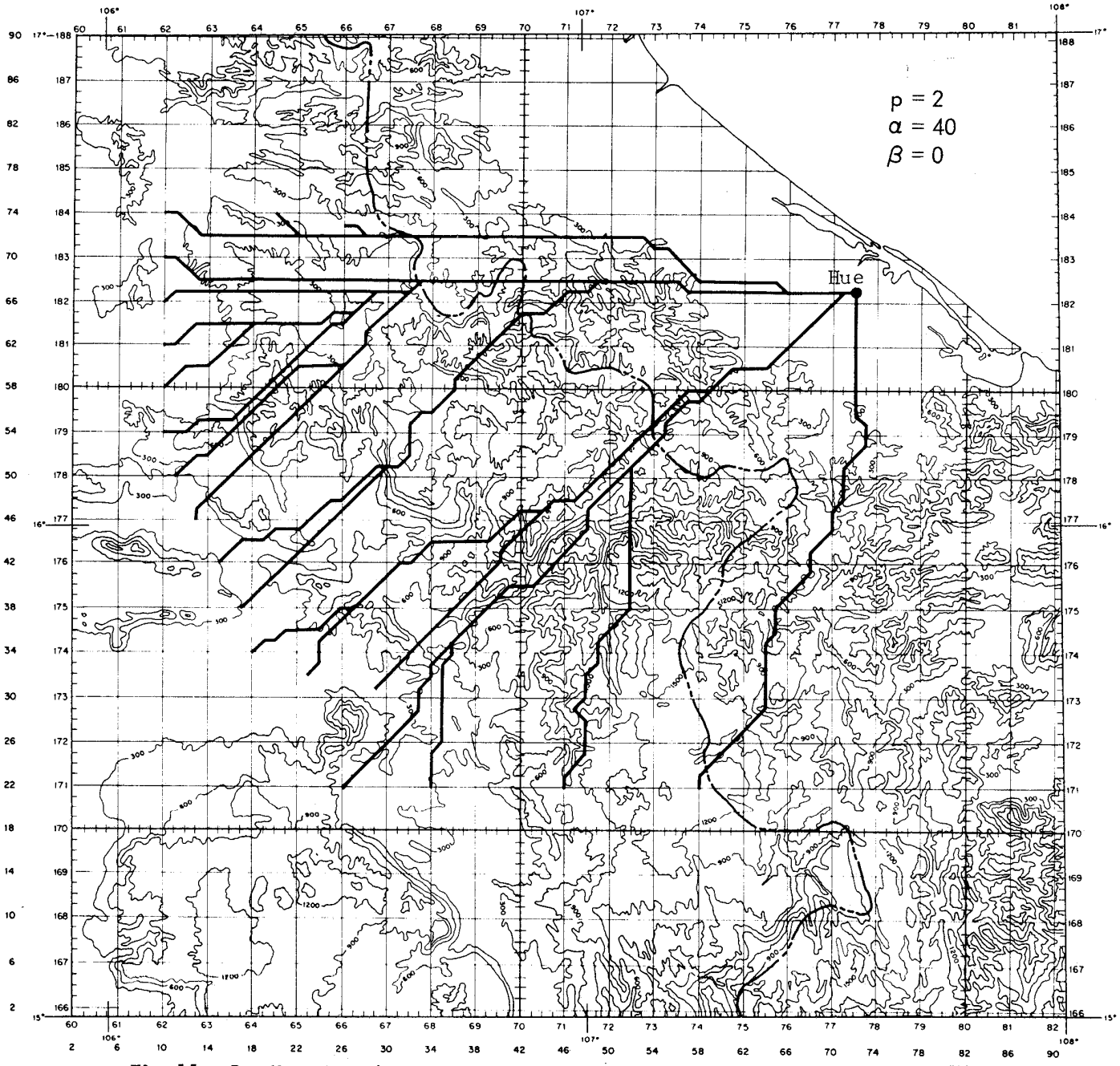


Fig.11—Predicted infiltration routes, with terminal point at coordinates (72, 67)
(p = 2, α = 40, β = 0)

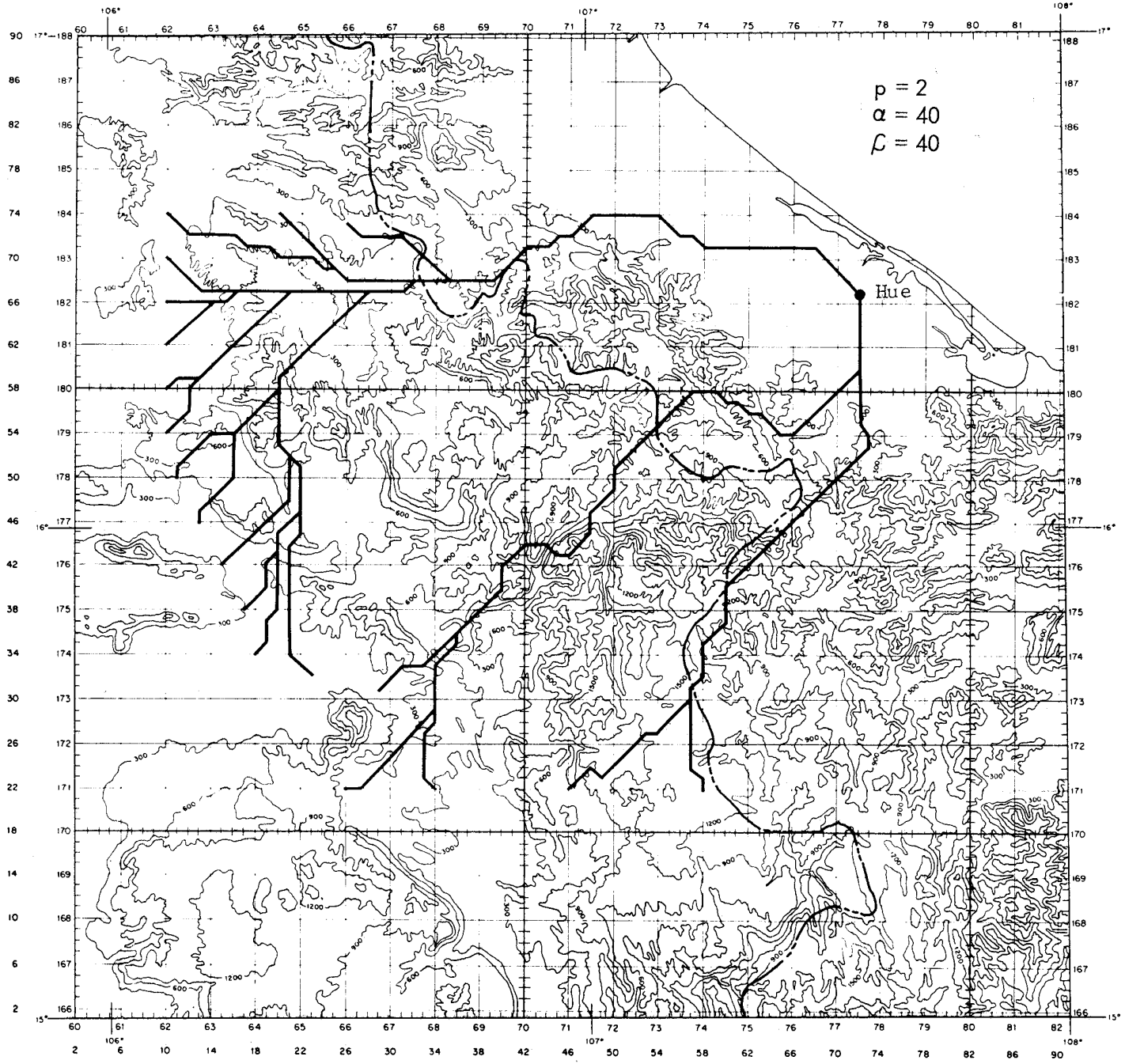


Fig. 12—Predicted infiltration routes, with terminal point at coordinates (72, 67)
(p = 2, α = 40, β = 40)

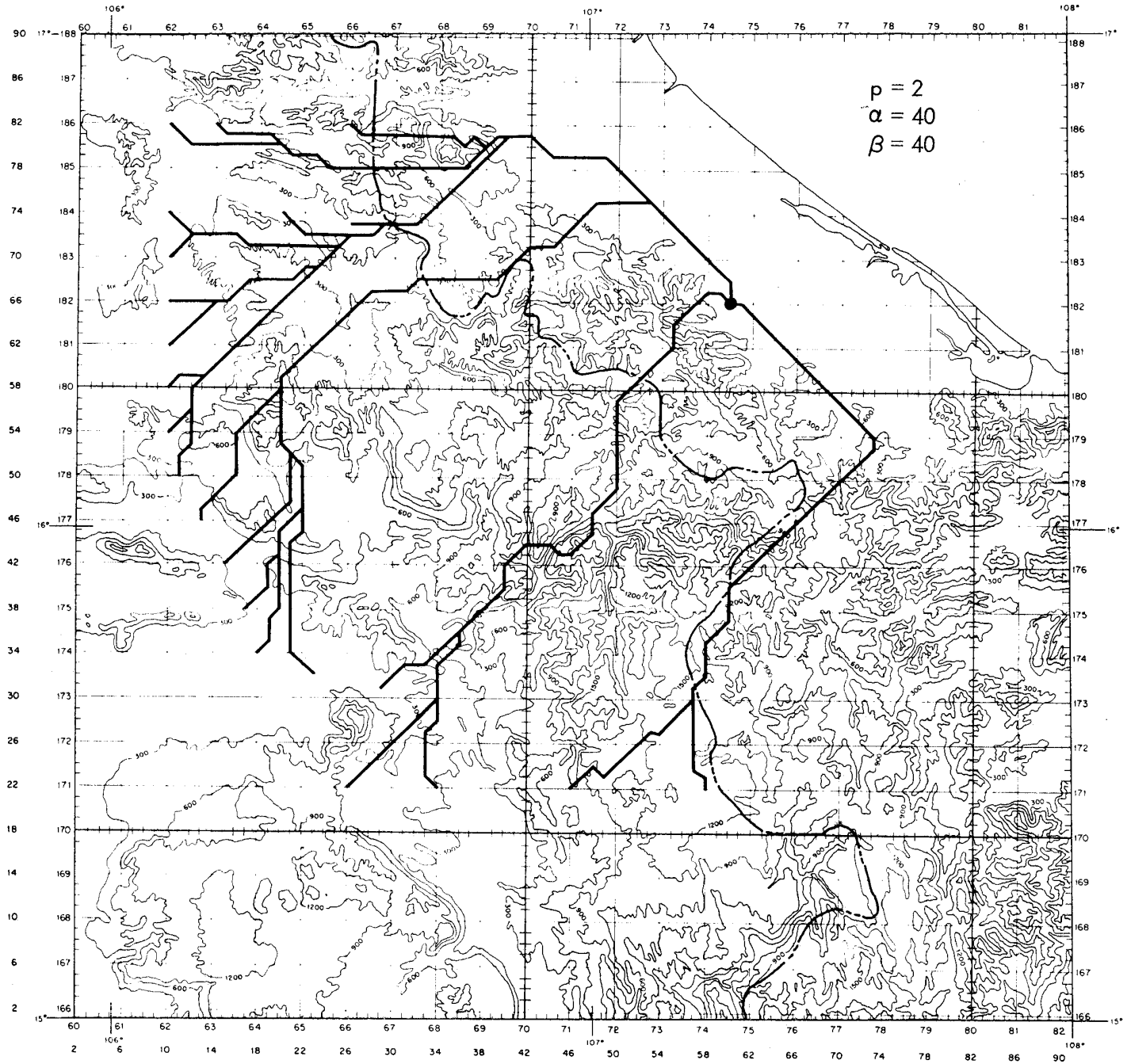


Fig. 13—Predicted infiltration routes, with terminal point at coordinates (60, 66) ($p = 2$, $\alpha = 40$, $\beta = 40$)

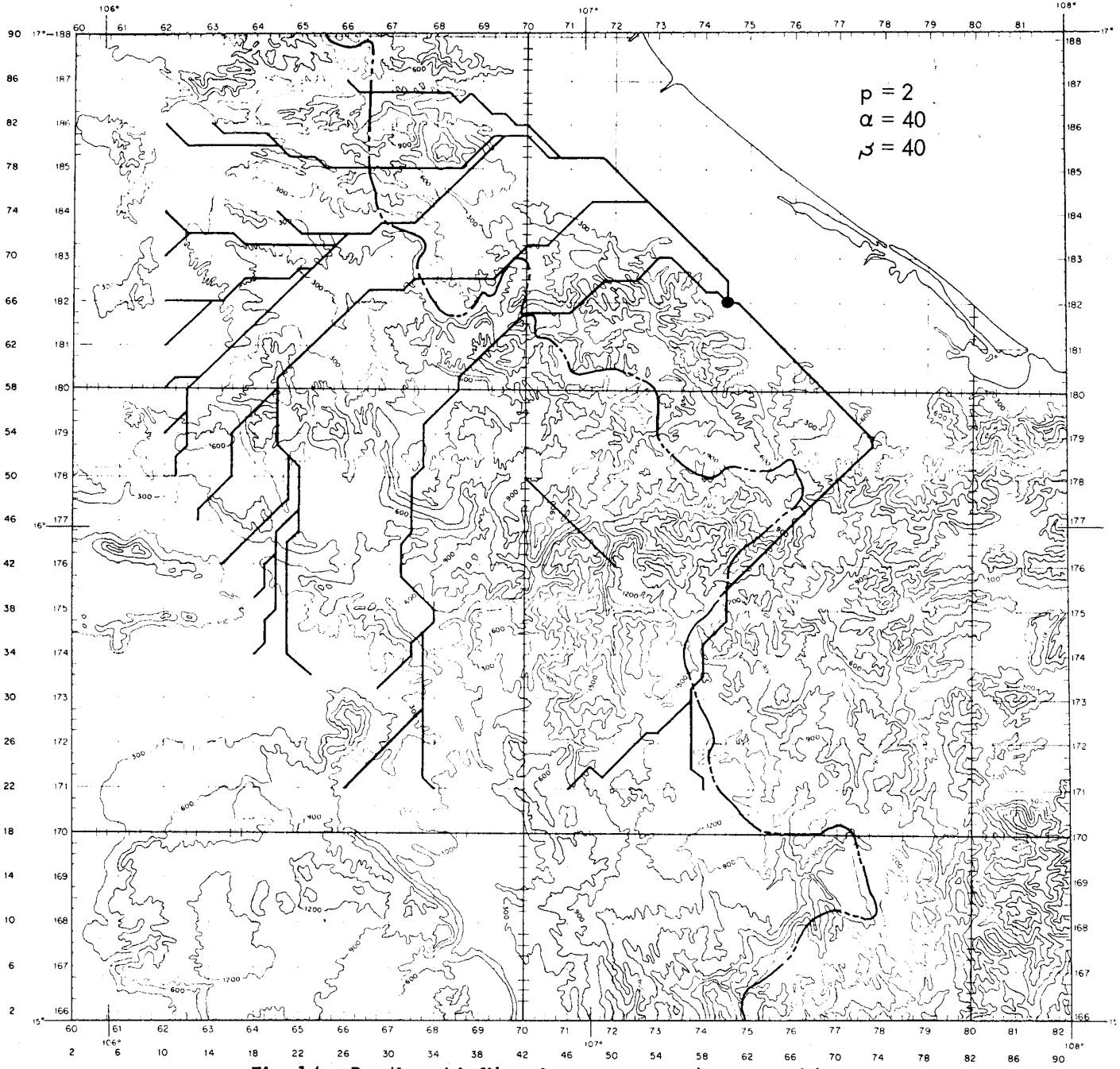


Fig.14—Predicted infiltration routes, with diagonal barrier and terminal point at coordinates (60, 66) (p = 2, α = 40, β = 40)

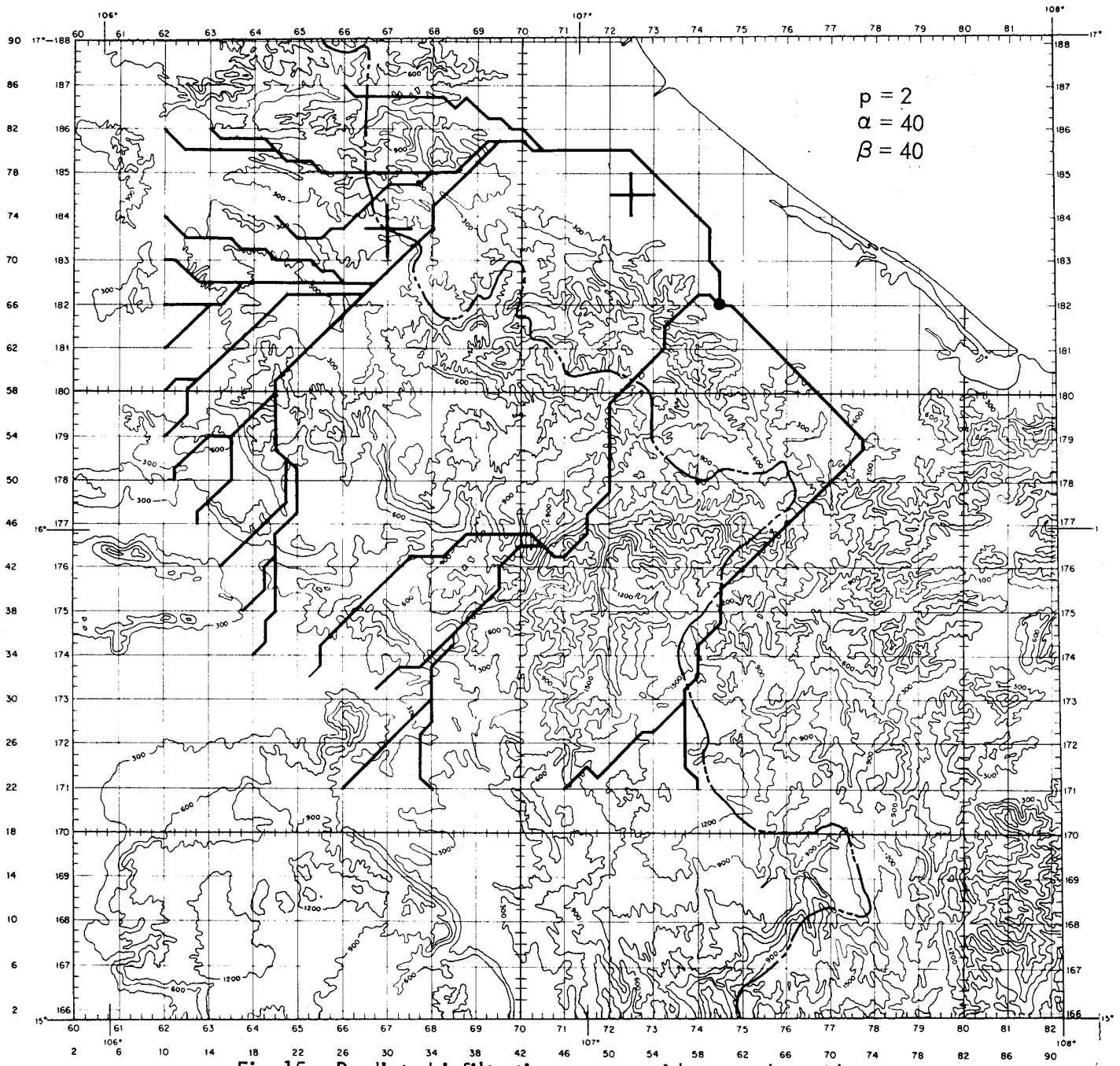


Fig. 15—Predicted infiltration routes, with cross-shaped barrier and terminal point at coordinates (60, 66) (p = 2, α = 40, β = 40)

VI. APPLICATIONS OF THE MODEL

INFILTRATION ROUTE CHARACTERISTICS

It has never been seriously suggested that infiltration routes are chosen at random. Traditional communication routes almost always follow easy rather than difficult terrain. Even if a route is discovered accidentally and further exploration stops, it will have some "shortest-path" characteristics even though the route may not be optimum. The model provides a systematic approach for investigating the criteria and degree of randomness involved in route determination, provided some infiltration routes are known to the investigator.

For example, if a given route appears to follow easy terrain, it is possible to use the model to predict various routes between end points of the known route, using different kinds of terrain cost functions. If the real route departs systematically from the "terrain optimal" routes in some region, an attempt would then be made to determine how the real and predicted routes in this region differ. Questions might be asked such as, Does the real route avoid villages that the predicted route approaches? Does the real route detour to follow a traditionally established path? Does the real route follow a path where vegetation provides cover from the air? Could the deviation be due to the existence of an unknown base camp or logistics camp? Once a reason for the deviation is conjectured, a new cost function may be constructed which includes a weighting factor for the new condition, and the process is repeated using new route predictions and looking for further departures from the predicted routes.

It is clear from this discussion that route characteristics other than the optimizing criteria can be inferred. For example, it may be possible to discover camps or rallying points by examining the deviation of real routes from those predicted. When fragmentary route information exists, it may be possible to determine start or end points for the real routes by matching the route fragments to routes predicted using an array of start and end points.

By use of sensors along known and predicted routes, it may be possible to determine shifts of traffic from one route to another; and by

use of the model, to determine whether the traffic shift is due to a change of objective (end point) or a change of selection criterion (for example, heavy logistics require easier terrain).

The alternate-route spectrum can also be investigated by the model. For each predicted route we can examine alternate routes which have a cost only slightly larger than that of the optimum route. In the case where small changes in the route result in large increases in route costs we can be confident the predicted route is significant. When many alternate routes exist at only slight cost increases, then the predicted route should not be taken too much for granted and the whole alternate-route array should be examined.

BARRIER AND SENSOR EMPLACEMENT

In many cases it may be necessary to construct barriers or strong points to prevent or restrict infiltration. Since the resources available to the infiltrator are usually limited, some kind of criterion for allocating resources for barrier construction is desirable. If the terrain is at all rugged, it would not seem wise to construct a barrier using the criterion of equal cost per mile of border. More resources should be expended where the cost to the infiltrator is otherwise small.

One approach to barrier resource allocation, for example, would be to prepare a route map from the model, such as that shown in Fig. 13, using a number of start points and one or more appropriate end points. Where several routes join to pass through choke points, barriers can be emplaced in the model as was done in Fig. 14, and the increased cost of the new route can be examined. If the increase is sufficiently large, a fairly substantial barrier would be indicated.

A more systematic approach would be to examine infiltration costs along the entire border (or barrier line) by choosing points along the border, first as terminal points from and then as start points for some point within the defended country, and adding the costs at each border point for the two routes leading to and away from the point. Since the barrier represents an additional cost to the infiltrator and this barrier penetration cost should increase as the cost of the barrier increases, then barrier resources can be allocated so that the total cost

to the infiltrator (infiltration plus barrier penetration) is the same for all points of border crossing.

An extension of this approach allows the determination of the optimum location of the barrier line. When the barrier along a given line is optimized, the infiltration cost is uniquely determined. Holding the resources for barrier emplacement fixed, and recognizing that barrier costs increase as the barrier is placed further into the mountains away from populated areas, we can determine where the barrier line for fixed total resources presents the highest total cost to the infiltrator.

This approach can also be used for the placement of sensor fields. It is assumed that the infiltrator would like to avoid sensor fields; increasing the density of these fields in regions of easiest travel, thereby causing the infiltrator to avoid these regions, will increase his costs.

Identification of choke points (i.e., points around which alternate routes have a large incremental cost) suggests positions for strong-point defenses. It was evident without analysis why Khe Sanh was a good location for restricting infiltration. Quite objectively, Fig. 13 points up distinctly the value of the Route 9 corridor. However, the model also indicates the width of the corridor and the range of routes over which infiltration costs are relatively stable. To be effective, a strong point at Khe Sanh would have to defend a very large area.

OFFENSE APPLICATIONS

The model should be of value to the infiltrator as well as the defender. It can recommend routes quickly in areas where the infiltrating force has little experience. Perhaps the most effective criticism of the model used by the defender is that the infiltrator may not be as systematic as the model implies. This problem is not important to the infiltrator using the model. He can estimate the costs of terrain for known personnel and equipment. He knows his start and end points. He knows how to weight population factors and detection by air, etc. Except for having to consider unknown hazards and factors which would be a problem in any case, the model can produce a number of preferred infiltration routes and estimates of the costs and time required to traverse each route.

Other applications of the model can be suggested; in many cases these are similar to techniques already being applied. Choosing a minimum-cost route to construct a road between two points is an obvious application. Finding the best routes for ground resupply of an advanced post is another. Finding the most effective ambush points when a sensor signals traffic at some distance is a possibility. There exists a range of applications such as determining boundaries of political subdivisions on the basis of ease of communication which is too broad for further consideration at this time.

Considerably greater experience with the model is required to fully estimate its capabilities and potentialities. Its concept is simple, but the diversity of parameters it encompasses provides a rich field for further investigation.

Appendix A

A JOSS PROGRAM FOR INVERSE-SQUARE INTERPOLATION

When data are available on a two-dimensional rectangular grid, it is often of interest to determine values and contour lines on a scale finer than the data grid by interpolation. There are many ways interpolation can be done, but we chose an inverse-square technique. We present here a simple JOSS program which can be adapted to other remote console systems with similar formatting.

The program listed below accepts 49 data inputs (from a 7 × 7 grid) as z(i, j). Parts 1, 2, and 3 input the data. Step 3.2 was added to avoid having to input a decimal point. Part 99 can be used to start the program if the data are already in storage.

The interpolated value at each point is calculated as a weighted average of the 16 nearest input grid values. The weighting function is proportional to the square of the reciprocal of the distance between the interpolation point and each of the 16 data points. The calculation proceeds by interpolating a horizontal line at the top of the area, printing that line, and then repeating the process for the next line down, and so on.

The interpolated area lies within the 5 × 5 subgrid of the original 7 × 7 data grid. We have chosen as an example an interpolation interval of one-tenth of the original data-grid interval.

Step 1.2 does the vertical scan of the program. Step 4.1 activates parts 5, 8, 9, and 7 to calculate the interpolation along a horizontal line. Each interpolated value is calculated as h(x) in Step 5.91.

Steps 4.2 and 4.3 then initiate an unlikely looking set of calculations in parts 10 and 11, which arise because of the formatting limitations of JOSS. Part 10 selects one digit to be printed at the interpolated point representing a truncated value of h(x). Part 11 combines five of these digits into a single number b(k), alternating the digits with the digit 1, which is a dummy. The purpose of the dummy is to space the digits horizontally equal to the vertical spacing of the printed lines. Since JOSS will not print a leading zero in a number, Step 11.3 was added to use a dash to avoid an empty space where a zero should be.

Steps 4.31, 4.32, and 4.33 do the same as part 11 except for combining four digits rather than five, since we could not fit the fifth digit into the final space of the form line. Step 4.4 does the final output for each horizontal line by typing the eight numbers, b(k), into the form, giving a continuous line of 39 digits separated by 38 dummy digits.

Figure A-1 shows an area of South Vietnam drawn from a map with 300-m contour lines. Figure A-2 is a sample output from the interpolation program of the same area, where each digit (not a dummy) represents the altitude in 100-m intervals. The data-input grid is the same as that shown in Fig. A-1, but grid points surrounding the interpolated area are not shown.

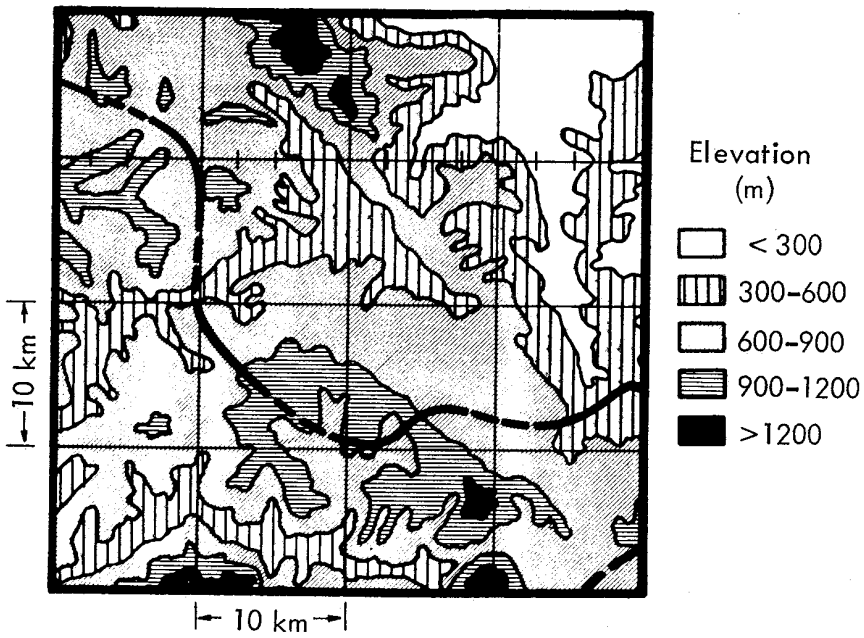


Fig.A-1—Map contours

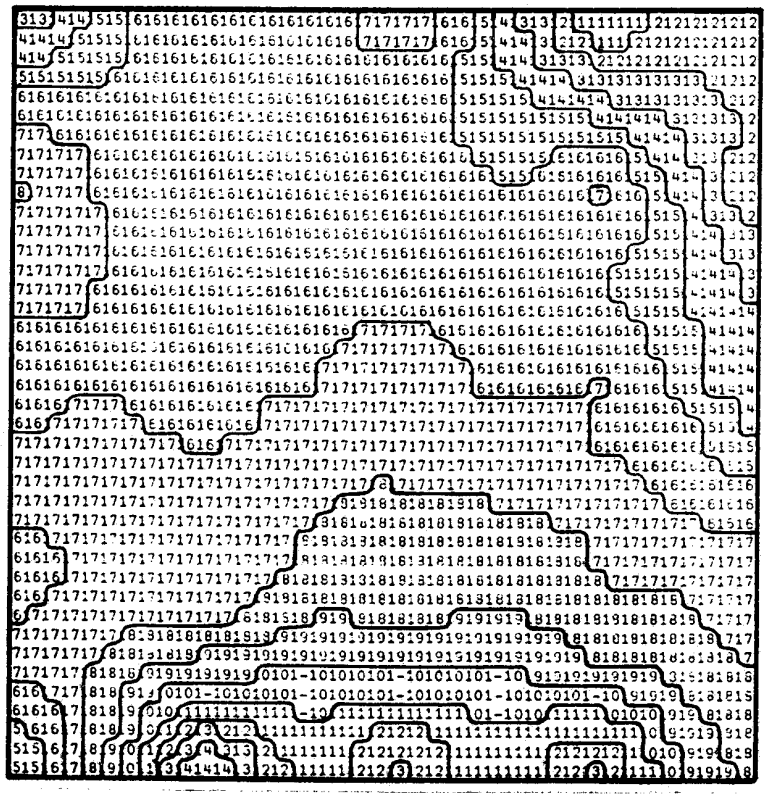


Fig.A-2—Interpolation

```
Use file 270 (35850).
Roger.
Recall item 25 (MAP).
Done.
Type all,size.

1.1 Do part 2 for j=5(-1)-1.
1.15 Page.
1.16 Set c=100.
1.2 Do part 4 for y=39(-1)0.

2.1 Do part 3 for i=-1(1)5.

3.1 Demand z(i,j).
3.2 Set z(i,j)=z(i,j)/10.

4.1 Do part 5 for x=0(1)39.
4.2 Do part 10 for x=0(1)39.
4.3 Do part 11 for k=0(5)30.
4.31 Set b(35)=a(38)+c•(a(37)+c•(a(36)+c•a(35))).
4.32 Set b(35)=b(35)+101010.
4.33 Set b(35)=-b(35) if a(35)=0.
4.4 Type b(0),b(5),b(10),b(15),b(20),b(25),b(30),b(35) in form 1.

5.1 Do part 8 if fp(x/10)=0 and fp(y/10)=0.
5.2 Done if fp(x/10)=0 and fp(y/10)=0.
5.3 Set N=0.
5.4 Set g(x)=0.
5.5 Set X= ip(x/10).
5.51 Set Y=ip(y/10).
5.9 Do part 9 for i=X-1(1)X+2.
5.91 Set h(x)=g(x)/N.

7.1 Set f(i,j)= (x/10-i)*2+(y/10-j)*2.
7.2 Set N=N+1/f(i,j).
7.3 Set g(x)=g(x)+z(i,j)/f(i,j).

8.10 Set m=x/10.
8.11 Set n=y/10.
8.2 Set h(x)= z(m,n).
8.3 Done.

9.1 Do part 7 for j=Y-1(1)Y+2.

10.1 Set a(x)=ip(10•fp(h(x)/10)).

11.1 Set b(k)=a(k+4)+c•(a(k+3)+c•(a(k+2)+c•(a(k+1)+c•a(k)))).
11.2 Set b(k)= b(k)+ 10101010.
11.3 Set b(k)=-b(k) if a(k)=0.

99 To step 1.15.

Form 1:
_____1_____1_____1_____1_____1_____1_____1_____

size =  218
```

Appendix B

PREPARATION OF AN INPUT DECK FOR THE FORTRAN IV VERSION OF PREMIR

INPUT DATA

Control Variables

1. NUMX — is the number of the last column of data in the input matrix (grid).
2. NUMY — is the number of the last row of data in the input matrix (grid).
3. NUMX2 — is the actual number of columns of data from the input matrix which will be retained in the subsequent network.
4. NUMY2 — is the actual number of rows of data from the input matrix which will be retained in the subsequent network.
5. NX — is the number of the first column of data in the input matrix.
6. NY — is the number of the first row of data in the input matrix.
7. NDØX — is the beginning horizontal (X) index for nodes in the subsequent network system.
8. NDØY — is the beginning vertical (Y) index for nodes in the subsequent network system.
9. INCX — is the incremental distance between adjacent nodes of the network which are horizontally separated.
10. INCY — is the incremental distance between adjacent nodes of the network which are vertically separated.
11. ICØST — indicates whether or not the input and interpolated data values and cost calculations are to be printed as output. If ICØST > 0, the information will be printed.
12. IBLØCK — is the number of nodes in the network to which travel from surrounding nodes is to be blocked.
13. INTER — indicates whether or not interpolation is to be applied. If INTER > 0, interpolation *will not* be applied.

14. NDEC — is the number of decimal places to which interpolated values are to be rounded.

15. IR — indicates whether or not the roughness factor is to be included in the cost calculations. Input IR > 0 for inclusion.

16. IGRIDX — is the number of interpolated data points +1 between adjacent input data points measured in the horizontal direction.

17. IGRIDY — is the number of interpolated data points +1 between adjacent input data points measured in the vertical direction.

Cost Data

1. ZERØ — is the altitude-limiting variable. All altitude values (input and interpolated) less than or equal to ZERØ will be considered undesirable for travel (e.g., the ocean), and a high cost will be assigned to them.

2. C, G, H, ALPHA, P, BETA — are parameters of the cost function (see Section IV). Input H as "1.0".

Dynamic Program Data

1. IDØX, ITØX — are the horizontal component (X) indexes of the boundary nodes (lower and upper, respectively) for the area of the network in which optimal routes are to be predicted.

2. JDØY, JTØY — are the vertical component (Y) indexes of the boundary nodes (lower and upper, respectively) for the area of the network in which optimal routes are to be predicted.

3. ISTART — is the number of iterations the model performs before testing for the optimal solution.

4. T(1), T(2) — are the X and Y coordinates (respectively) of the node designated as the terminal.

CARD INPUT FØRMAT

A denotes alphanumeric data.

I denotes integer data right-adjusted in its field.

F.d denotes real numeric data with d = number of decimal points.

First Card

Col	1-72	Code name for run (may be left blank).	

Matrix and Network Data

If interpolation is applied, then the outside bordering input data points may not be included as nodes in the network system, since an interpolated data point requires that a somewhat symmetric neighborhood of input data points be used. The input data are considered as a mxn matrix and the user inputs to the model information relating to the form and structure of the matrix. The resulting network is rectangular in shape with each node having x, y coordinates.

Second Card

Col	1-5	Last column number of input data matrix (NUMX)	(I)
Col	6-10	Last row number of input data matrix (NUMY)	(I)
Col	11-15	Number of data columns retained in network (NUMX2)	(I)
Col	16-20	Number of data rows retained in network (NUMY2)	(I)
Col	21-25	First column number (NX)	(I)
Col	26-30	First row number (NY)	(I)
Col	31-35	Beginning X index for network (NDØX)	(I)
Col	36-40	Beginning Y index for network (NDØY)	(I)
Col	41-45	Horizontal incremental distance (INCX)	(I)
Col	46-50	Vertical incremental distance (INCY)	(I)

Third Card

Col	1-5	Print mode (ICØST)	(I)
Col	6-10	Number of blocked nodes (IBLØCK)	(I)
Col	11-15	Indicator for applying interpolation (INTER)	(I)
Col	16-20	Number of decimal places (NDEC)	(I)
Col	21-25	Indicator for including roughness factor (IR)	(I)
Col	26-30	Number of interpolated data points +1 measured horizontally (IGRIDX)	(I)
Col	31-35	Number of interpolated data points +1 measured vertically (IGRIDY)	(I)

Fourth Card

Col 1-5	Below-sea-level altitudes (ZERØ)		(F)
Col 6-10	C	Cost function parameters	(F)
Col 11-15	G		(F)
Col 16-20	H		(F)
Col 21-25	ALPHA		(F)
Col 26-30	P		(F)
Col 31-35	BETA		(F)

Grid Altitude Input Data

For convenience, the model is programmed to accept data in different formats as input through the use of a variable-format statement. Fortran provides for variable-format statements by allowing a format specification to be read into an array in storage and using the information in the array as the format specification for subsequent input/output. Thus, the user is given much flexibility in the form of the data inputs.

Fifth Card

Col 1-72 Variable format specification of the input data (FØRMT) (A)

The general form of the data input is given below:

$$\begin{matrix} D_{11} & D_{12} & \cdots & D_{1n} \\ D_{21} & D_{22} & \cdots & D_{2n} \\ \vdots & \vdots & & \vdots \\ D_{m1} & D_{m2} & \cdots & D_{mn} \end{matrix}$$

The entries D_{ij} in the table are the data values, and the whole array of these numeric values is called the input data matrix.

The general data-card input is keypunched row-wise. That is, all of the data values of the first row of the input matrix are punched in order on one or more cards. Then starting on a new card, the second row is punched, etc. The data are read into the model in a similar fashion.

Sixth Cards

Col	As specified by the variable-format	Input data values (D_{ij})

Seventh Card*

Col	1-3	X coordinate of first node to be blocked	(I)
Col	4-6	Y coordinate of first node to be blocked	(I)
Col	7-9	X coordinate of second node to be blocked	(I)
Col	10-12	Y coordinate of second node to be blocked	(I)
.....			
Col	54-57	X coordinate of tenth node to be blocked	(I)
Col	58-60	Y coordinate of tenth node to be blocked	(I)

If there are more than 10 nodes to be blocked, continue on another card of the same form, until IBLØCK nodes have been specified.

Dynamic Program Data

After creation of the network system, the model will predict either optimal routes for the entire network or some subset thereof, as specified by the user. This option is an economy consideration in that the user may want predicted routes for different areas of the network. An 8-order card, defining the boundaries, is required for each area.

Eighth Card

Col	1-5	X coordinate of lower bound (IDØX)	(I)
Col	6-10	X coordinate of upper bound (ITØX)	(I)
Col	11-15	Y coordinate of lower bound (JDØY)	(I)
Col	16-20	Y coordinate of upper bound (JTØY)	(I)
Col	21-25	Number of iterations before test (ISTART)	(I)
Col	26-30	X coordinate of terminal node	(I)
Col	31-35	Y coordinate of terminal node	(I)

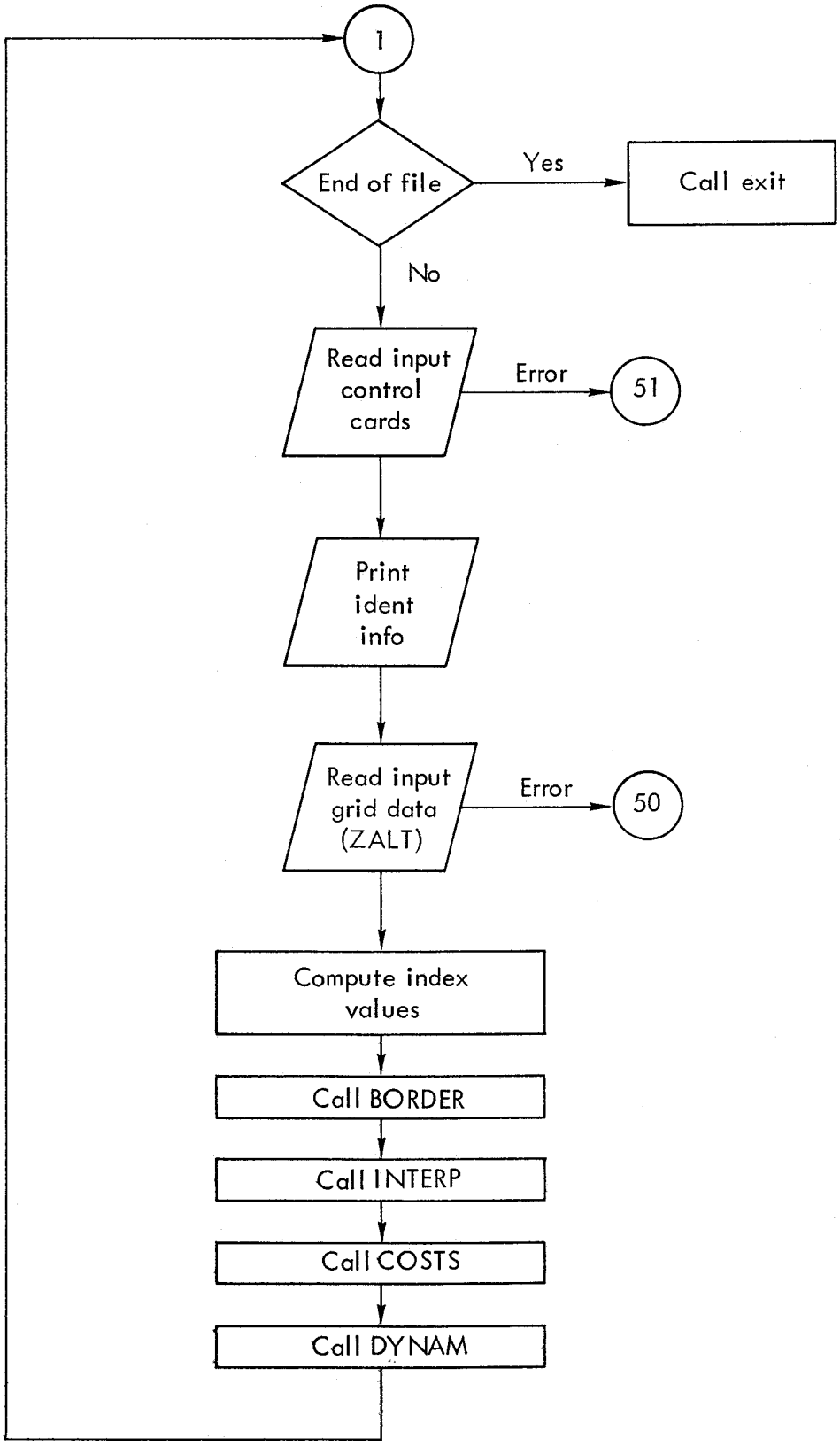
*Include only if IBLØCK > 0.

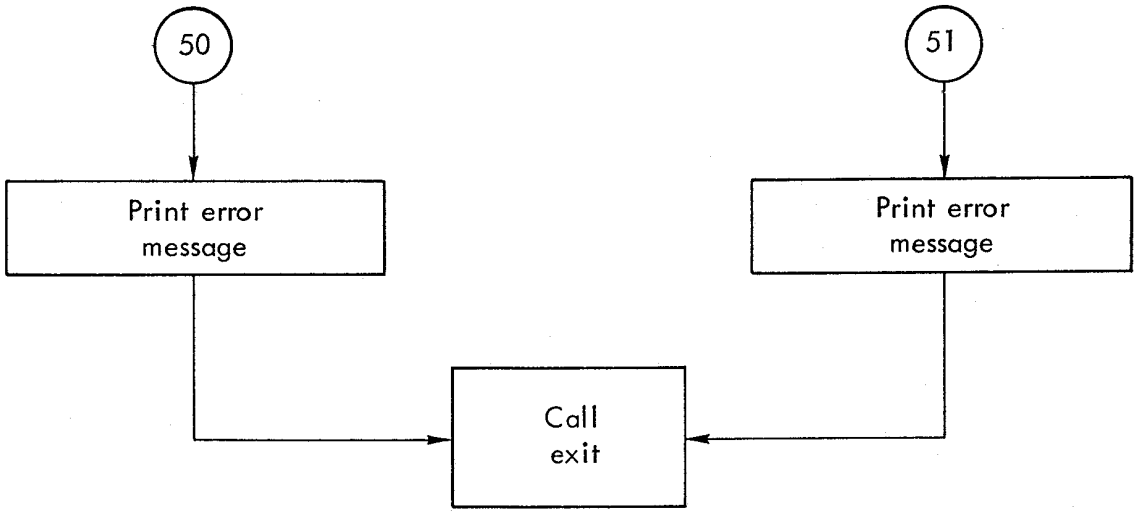
As many sets of cards 1-8 as there are cases involving different cost calculations may be included during a computer run, with each set separated by a blank card. The end of an input deck is specified by an end-of-file card.

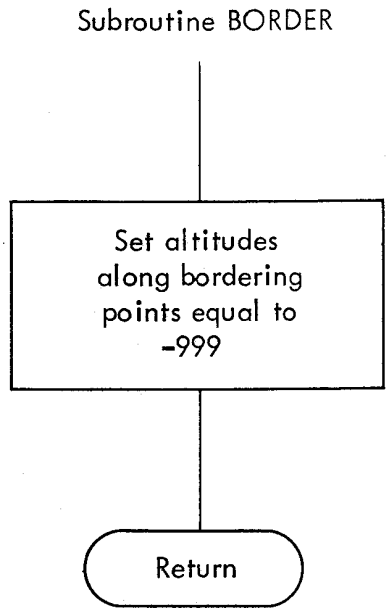
Appendix C

FLOW CHARTS OF THE PRINCIPAL SUBROUTINES TO PREMIR

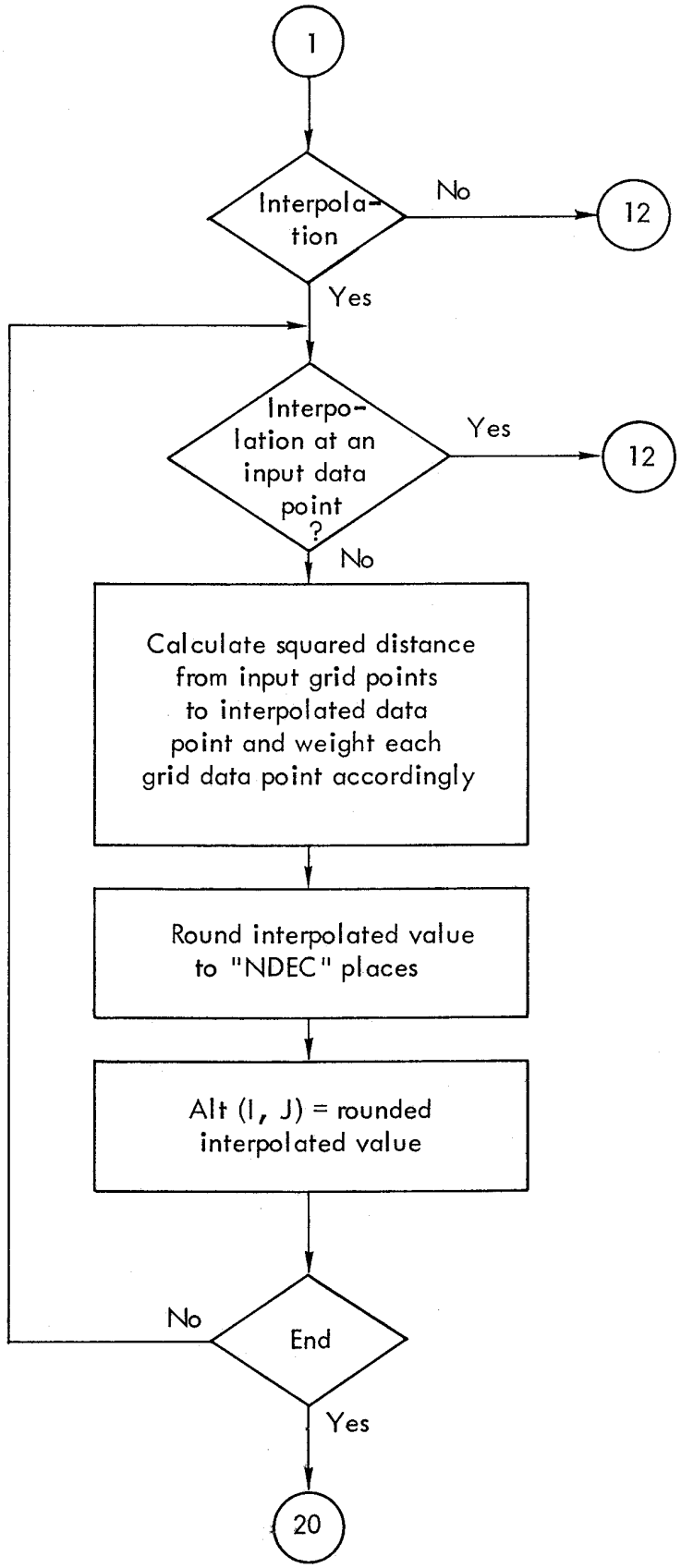
Main Program

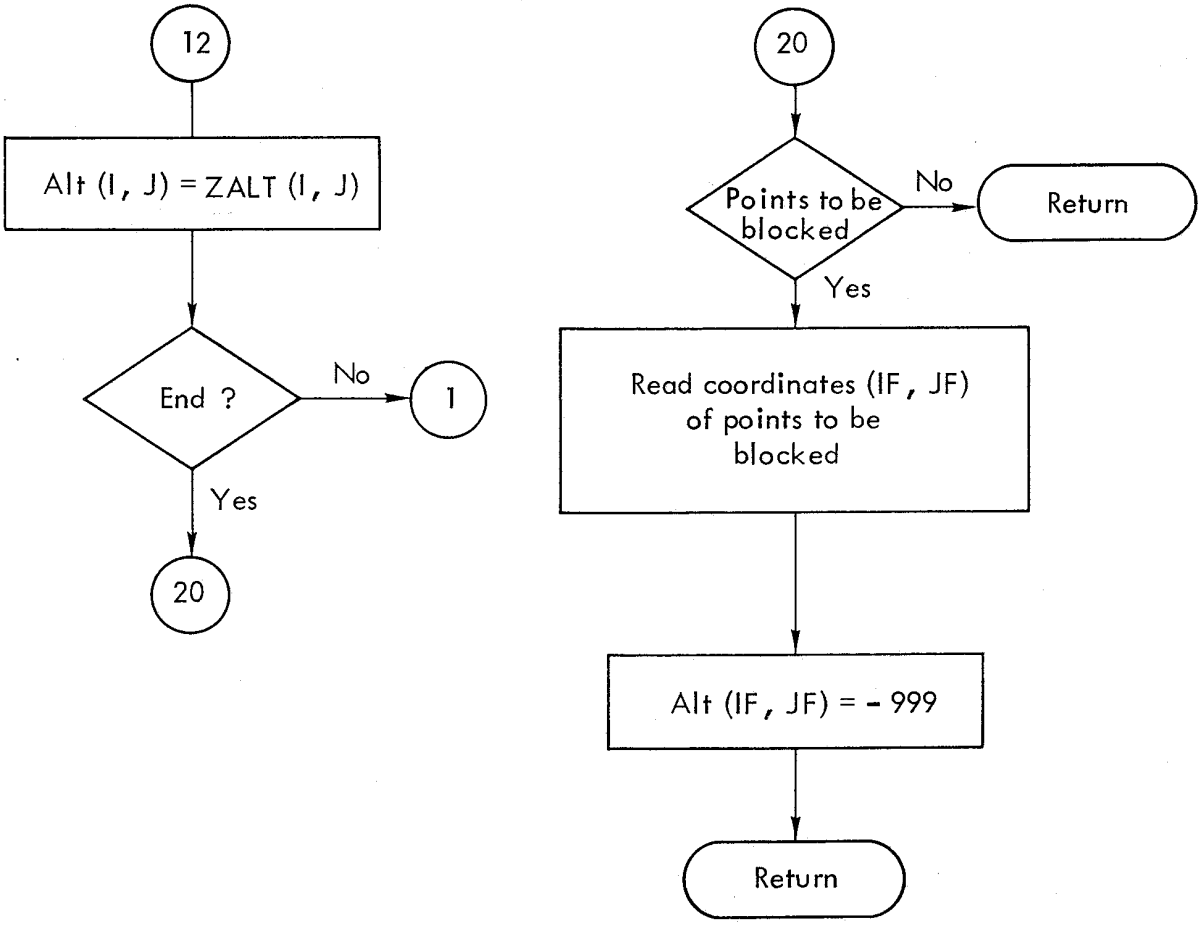



50
51
Print error message
Print error message
Call exit

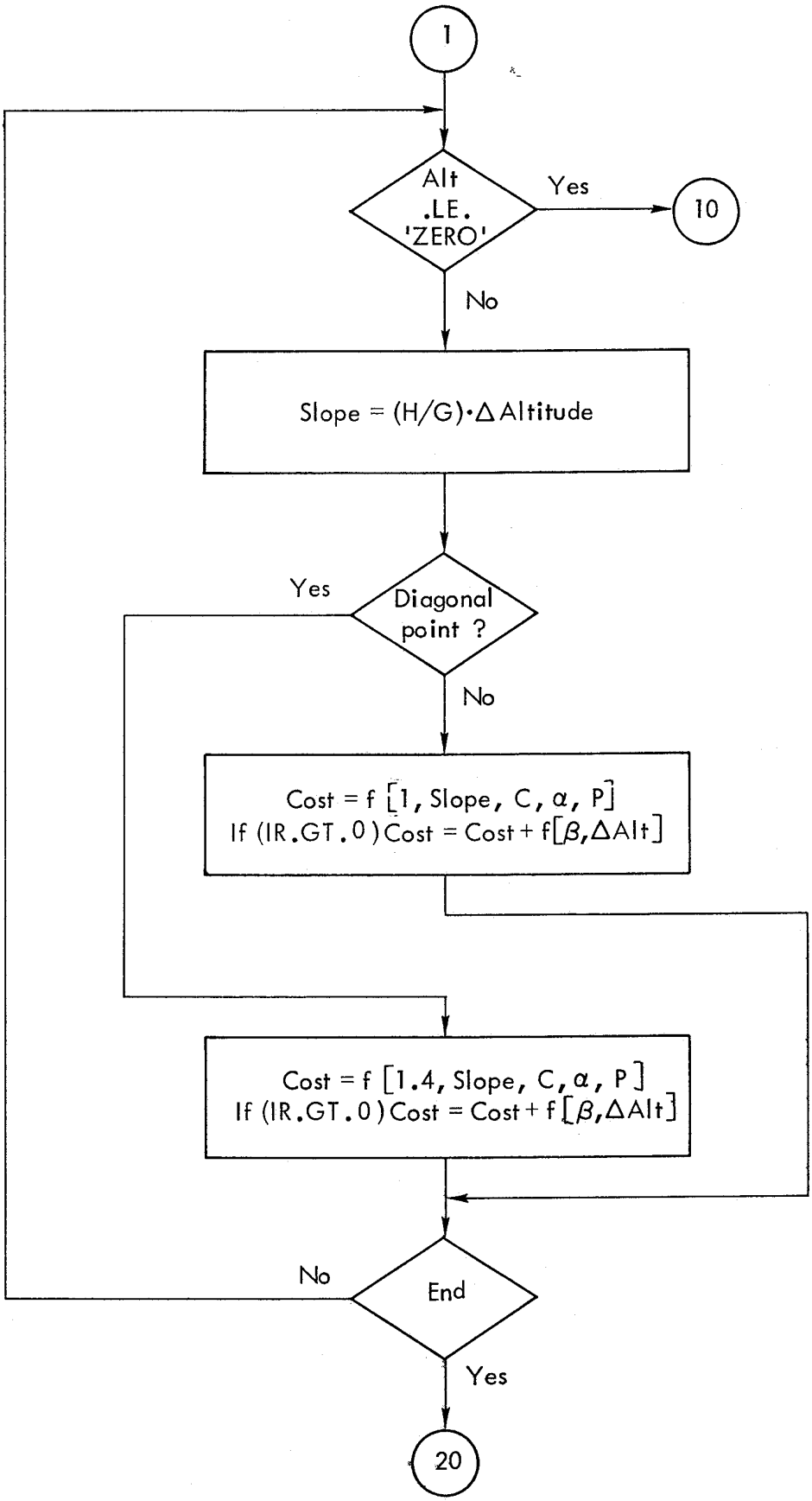

Subroutine BORDER
Set altitudes along bordering points equal to -999
Return

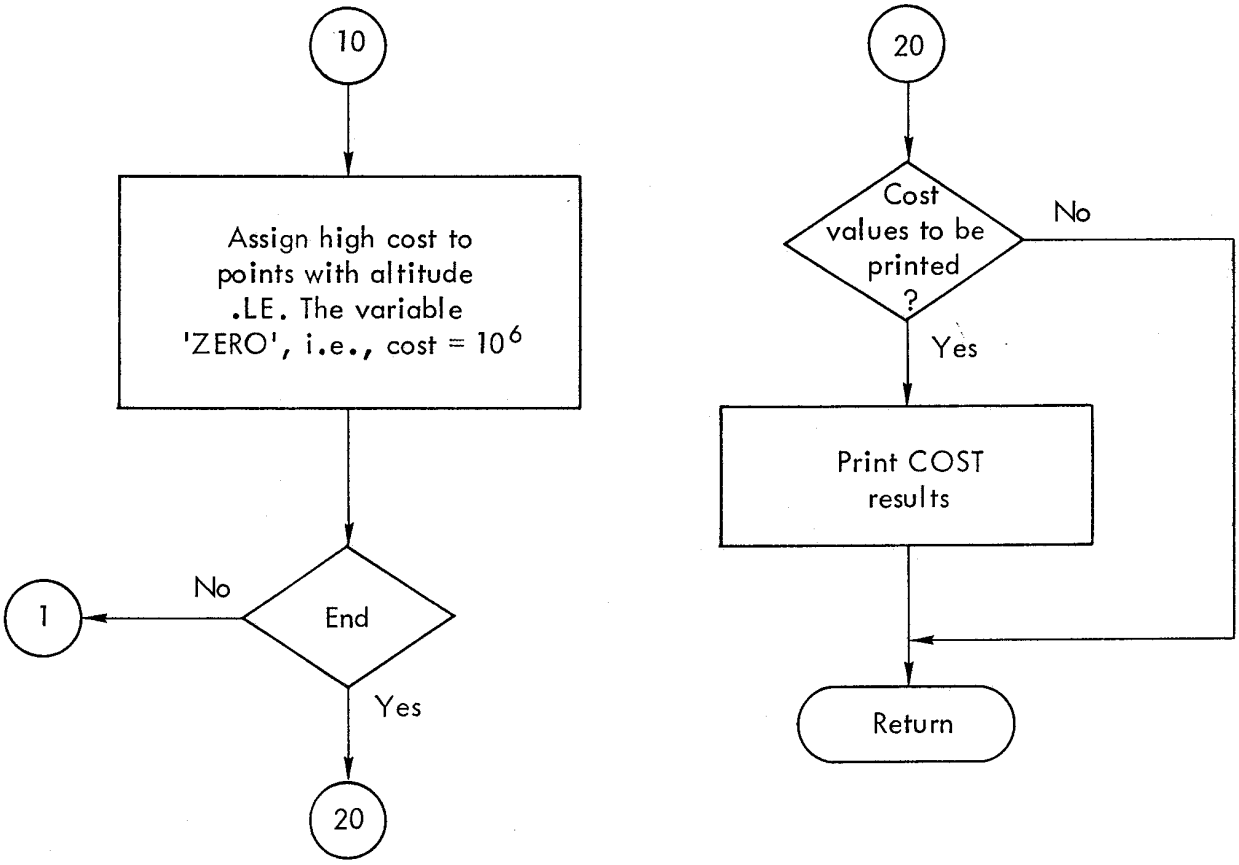
Subroutine INTERP



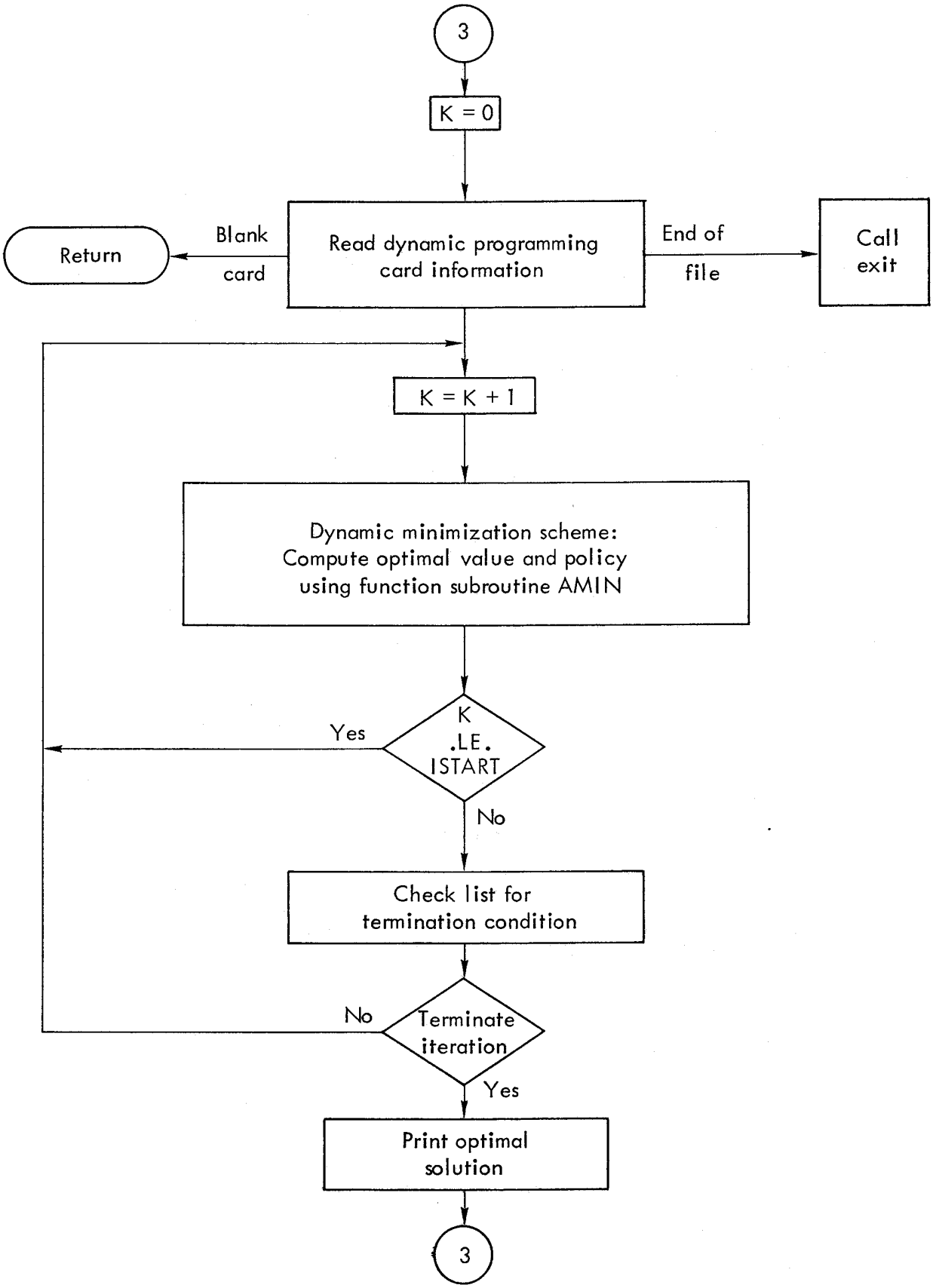

12
Alt (I, J) = ZALT (I, J)
End ?
No
1
Yes
20
20
Points to be blocked
No
Return
Yes
Read coordinates (IF, JF) of points to be blocked
Alt (IF, JF) = - 999
Return

Subroutine COST

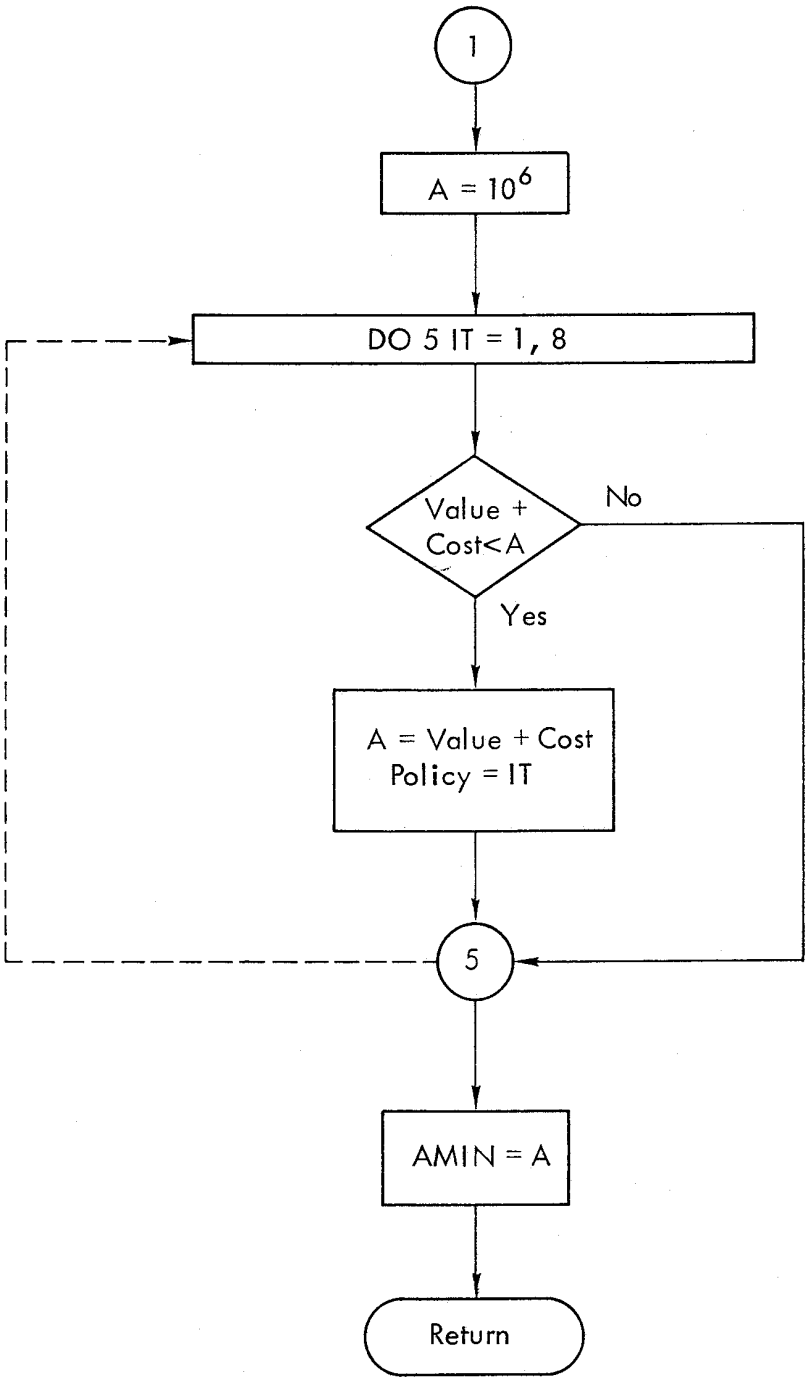



10
Assign high cost to points with altitude .LE. The variable 'ZERO', i.e., cost = 10^6
End
No
1
Yes
20
20
Cost values to be printed ?
No
Yes
Print COST results
Return

Subroutine DYNAMI



Function Subroutine AMIN



Appendix D

FORTRAN IV SOURCE LISTING OF PREMIR

```
      COMMON /ARRAYS/ COST(100,100,8),VALUE(100,100,2),T(2)
      COMMON /VALUES/ NTO1X,NTO1Y,NTOX,NTOY,NDOX,NDOY,INCX,INCY,IOLD,
     1I,J,IJJ,NUMX,NUMY,ICOST,IBLOCK,NDEC,IGRIDX,IGRIDY,ZERO,C,G,H,
     2ISTART,ALPHA,P,BETA,IR,INTER
      DIMENSION ZALT(100,100), ALT(100,100),FORMT(18),ANAME(18)
      EQUIVALENCE (ZALT(1),COST(1)), (ALT(1),VALUE(1))
C
C     INPUT PROBLEM CODE NAME
C
    1 READ(5,503,END=60) (ANAME(K),K=1,18)
      WRITE(6,506) (ANAME(K),K=1,18)
C
C     INPUT MODEL CONTROL DATA
C
      IERROR=2
      READ(5,500,ERR=51) NUMX,NUMY,NUMX2,NUMY2,NI,NJ,NDOX,NDOY,INCX,INCY
      IERROR=3
      READ(5,500,ERR=51) ICOST,IBLOCK,INTER,NDEC,IR,IGRIDX,IGRIDY
      IF(IR.GT.0) WRITE(6,600)
C
C     INPUT COST DATA
C
      IERROR=4
      READ(5,502,ERR=51) ZERO,C,G,H,ALPHA,P,BETA
      IERROR=5
C
C     INPUT GRID ALTITUDE DATA FORMAT
C
      READ(5,503) (FORMT(I),I=1,18)
      WRITE(6,507) (FORMT(I),I=1,18)
C
C     INPUT GRID ALTITUDE DATA INTO ARRAY ZALT
C
      DO 10 J=NJ,NUMY
      READ(5,FORMT,ERR=50) (ZALT(I,J),I=NI,NUMX)
   10 CONTINUE
C
C     THE FOLLOWING COMPUTED VALUES ARE USED IN SUBSEQUENT ROUTINES
C     FOR CONTROLLING DO-LOOPS, ETC.
C
      NTOX=IGRIDX*(NUMX2-1) + NDOX
      NTOY=IGRIDY*(NUMY2-1) + NDOY
      NTO1X=NTOX+INCX
      NTO1Y=NTOY+ INCY
C
C     THE FOLLOWING CALL IS FOR SETTING UP BORDER CONDITIONS
C
      CALL BORDER
C
C     THE FOLLOWING CALL IS FOR INTERPOLATING BETWEEN INPUT GRID POINTS
C
      CALL INTERP
C
C     THE FOLLOWING CALL IS FOR COMPUTING THE COSTS OF TRAVELING TO
C     ADJACENT POINTS AND ADJUSTING FOR BLOCKED NODES.
C
```

```
      CALL COSTS
C
C     THE FOLLOWING CALL IS TO THE DYNAMIC RECURSION ROUTINE
C
      CALL DYNAMI
      GO TO 1
   50 WRITE(6,504) I,J
      CALL EXIT
   51 WRITE(6,505) IERROR
   60 CALL EXIT
      STOP
  500 FORMAT(16I5)
  501 FORMAT(16F5.2)
  502 FORMAT(16F5.0)
  503 FORMAT(18A4)
  504 FORMAT('1','ERROR OCCURRED WHILE READING INPUT GRID DATA.  COORDIN
     1ATE VALUES: X=',I4,' Y=',I4)
  505 FORMAT('1',5X,'ERROR OCCURRED WHILE READING INPUT CARD NO.',I3,'.'
     1)
  506 FORMAT ('1',2X,'PROBLEM CODE:',18A4)
  507 FORMAT('0',2X,'INPUT FORMAT SPECIFICATION:',2X,18A4)
  600 FORMAT('0',2X,'R FACTOR INCLUDED IN COST CALCULATIONS.')
      END
      SUBROUTINE BORDER
      COMMON /ARRAYS/ COST(100,100,5),VALUE(100,100,2),T(2)
      COMMON /VALUES/ NTO1X,NTO1Y,NTOX,NTOY,NDOX,NDOY,INCX,INCY,IOLD,
     1I,J,IJJ,NUMX,NUMY,ICOST,IBLOCK,NDEC,IGRIDX,IGRIDY,ZERO,C,G,H,
     2ALPHA,P,BETA,IR
      DIMENSION ZALT(100,100), ALT(100,100)
      EQUIVALENCE (ZALT(1),COST(1)), (ALT(1),VALUE(1))
C
C     SUBROUTINE ASSIGNS SPECIAL ALTITUDES TO BORDER POINTS
C     NTO1X IS THE UPPER LIMIT IN THE X DIRECTION
C     NTO1Y IS THE UPPER LIMIT IN THE Y DIRECTION
C
      NI=NDOX-INCX
      NJ=NDOY-INCY
      DO 5 I=NI,NTO1X,INCX
      ALT(I,NJ)=-999
      ALT(I,NTO1Y)=-999
    5 CONTINUE
      DO 10 J=NJ,NTO1Y,INCY
      ALT(NI,J)=-999
      ALT(NTO1X,J)=-999
   10 CONTINUE
      RETURN
      END
      SUBROUTINE INTERP
      COMMON /ARRAYS/ COST(100,100,8),VALUE(100,100,2),T(2)
      COMMON /VALUES/ NTO1X,NTO1Y,NTOX,NTOY,NDOX,NDOY,INCX,INCY,IOLD,
     1I,J,IJJ,NUMX,NUMY,ICOST,IBLOCK,NDEC,IGRIDX,IGRIDY,ZERO,C,G,H,
     2ISTART,ALPHA,P,BETA,IR,INTER
      DIMENSION IF(10),JF(10)
      DIMENSION ZALT(100,100), ALT(100,100)
      EQUIVALENCE (ZALT(1),COST(1)), (ALT(1),VALUE(1))
C
C     INTERPOLATION SCHEME
C
      DO 20 I=NDOX,NTOX,INCX
```

```
      DO 20 J=NDOY,NTOY,INCY
      IF(INTER.GT.0) GO TO 11
      A=(I-2.*INCX)/IGRIDX
      B=(J-2.*INCY)/IGRIDY
      IA=A
      IB=B
      A=A-IA
      B=B-IB
      IF(A+B.EQ.0) GO TO 12
      GX=0
      DX=0
      NST1=IA+1
      NST2=IB+1
      NST3=NST1+3
      NST4=NST2+3
      IF(NST3.GT.NUMX) NST3=NUMX
      IF(NST4.GT.NUMY) NST4=NUMY
C
C     CALCULATE SQ. DIST. FROM INPUT GRID POINTS TO INTERPOLATION
C     POINTS AND WEIGHT EACH GRID DATA POINT ACCORDINGLY
C
      DO 10 IX=NST1,NST3
      DO 10 IY=NST2,NST4
      DIST2=((I-2.*INCX)-IGRIDX*(IX-2.))**2.+((J-2.*INCY)-IGRIDY*(IY-2.)
     A)**2.
      DX= DX+ 1/DIST2
      GX= GX+ ZALT(IX,IY)/DIST2
   10 CONTINUE
C
C     USE 'IQ' TO ROUND OFF INTERPOLATED VALUE TO 'NDEC' DECIMAL POINTS.
C
      IQ=( GX/DX +(5.)*(10.)**(-( NDEC+1 ) ) )*10**NDEC
      ALT(I,J)= IQ/(10.0)**NDEC
      GO TO 20
   11 IA=I-2
      IB=J-2
C
C     INTERPOLATION IS AT AN INPUT DATA POINT OR INTERPOLATION IS NOT
C     BEING USED.
C
   12 ALT(I,J)= ZALT(IA+2,IB+2)
   20 CONTINUE
      IF(IBLOCK.LE.0) GO TO 30
      IFLAG=0
      B1=IBLOCK/10.
      IB=B1
      B2=B1-IB
      NUM=B1
      IF(B2.GT.0) NUM=NUM+1
   22 IFLAG=IFLAG+1
      READ(5,500) (IF(K),JF(K),K=1,10)
      K=10*B2
      IF((B1.GE.1).AND.(IFLAG.LE.B1)) K=10
      DO 25 IK=1,K
      ALT(IF(IK),JF(IK))=-999
   25 CONTINUE
      IF(IFLAG.LT.NUM) GO TO 22
   30 RETURN
  500 FORMAT(20I3)
```

```
      END
      SUBROUTINE COSTS
      COMMON /ARRAYS/ COST(100,100,8),VALUE(100,100,2),T(2)
      COMMON /VALUES/ NTOIX,NTOIY,NTOX,NTOY,NDOX,NDOY,INCX,INCY,IOLD,
     1I,J,IJJ,NUMX,NUMY,ICOST,IBLOCK,NDEC,IGRIDX,IGRIDY,ZERO,C,G,H,
     2ISTART,ALPHA,P,BETA,IR,INTER
      DIMENSION ZALT(100,100), ALT(100,100)
      EQUIVALENCE (ZALT(1),COST(1)), (ALT(1),VALUE(1))
      DO 20 I=NDOX,NTOX,INCX
      DO 20 J=NDOY,NTOY,INCY
C
C     EIGHT WAYS OUT OF EACH I,J POINT
C
      DO 20 IJOUT=1,8
      ITO=I+GOTOX(IJOUT)
      JTO=J+GOTOY(IJOUT)
      IF( ALT(ITO,JTO).LE.ZERO) GO TO 10
      A=IJOUT/2.
      IA=A
      A=A-IA
      SLOPE=(H/G)*( ALT(ITO,JTO)-ALT(I,J) )
      IF(A.EQ.0) GO TO 5
C
C     COST CALCULATION FOR TRAVELING TO A NON-DIAGONAL POINT
C
      COST(I,J,IJOUT)= ( 1. + C*(ABS(ALPHA*SLOPE))**P )
      IF(IR.GT.0) COST(I,J,IJOUT)=COST(I,J,IJOUT) + (BETA*(ALT(ITO+1,JTO
     1)-ALT(ITO-1,JTO))/(2*G))**2 + (BETA*(ALT(ITO,JTO+1)-ALT(ITO,JTO-1)
     2)/(2*G))**2
      GO TO 20
C
C     COST CALCULATION FOR TRAVELING TO A DIAGONAL POINT
C
    5 COST(I,J,IJOUT)= ( 1.4 + C*(ABS(.7*ALPHA*SLOPE))**P)
      IF(IR.GT.0) COST(I,J,IJOUT)=COST(I,J,IJOUT) + (BETA*(ALT(ITO+1,JTO
     1)-ALT(ITO-1,JTO))/(2*G))**2 + (BETA*(ALT(ITO,JTO+1)-ALT(ITO,JTO-1)
     2)/(2*G))**2
      GO TO 20
C
C     ATTATCH A HIGH COST TO POINTS WITH ALTITUDES LESS THAN OR EQUAL TO
C     THE VARIABLE 'ZERO'.
C
   10 COST(I,J,IJOUT)=10.**6
   20 CONTINUE
C
C     IF 'ICOST' IS GREATER THAN ZERO, PRINT OUT COST CALCULATIONS
C
      IF(ICOST.LE.0) GO TO 40
      CALL PAGE(LINE)
      DO 30 I=NDOX,NTOX,INCX
      DO 30 J=NDOY,NTOY,INCY
      LINE=LINE+1
      IF(LINE.GT.55) CALL PAGE(LINE)
      WRITE(6,501) I,J,ALT(I,J),(COST(I,J,IJ),IJ=1,8)
   30 CONTINUE
   40 RETURN
  501 FORMAT(1X,I4,I5,F9.1,9X,8G11.6)
      END
      SUBROUTINE DYNAMI
```

```
      COMMON /ARRAYS/ COST(100,100,8),VALUE(100,100,2),T(2)
      COMMON /VALUES/ NTO1X,NTO1Y,NTOX,NTOY,NDOX,NDOY,INCX,INCY,IOLD,
     1I,J,IJJ,NUMX,NUMY,ICOST,IBLOCK,NDEC,IGRIDX,IGRIDY,ZERO,C,G,H,
     2ISTART,ALPHA,P,BETA,IR,INTER
      DIMENSION POLICY(100,100)
      INTEGER T
      INTEGER*2 POLICY
    3 READ(5,515,END=60) IDOX,ITOX,JDOY,JTOY,ISTART,T(1),T(2)
      IF(IDOX.LE.0) RETURN
      IF(ISTART.LE.0) ISTART=1
      DO 5 IT=1,2
      DO 5 I=IDOX,ITOX
      DO 5 J=JDOY,JTOY
      VALUE(I,J,IT)=0
    5 CONTINUE
C
C     'K' IS THE ITERATION NUMBER
C
      K=0
    6 K=K+1
      CALL FIX(INEW,IOLD,K)
C
C     DYNAMIC MINIMIZATION SCHEME
C
      DO 10 I=IDOX,ITOX,INCX
      DO 10 J=JDOY,JTOY,INCY
      IF((I.EQ.T(1)).AND.(J.EQ.T(2))) GO TO 10
      VALUE(I,J,INEW)=AMIN(IDOX,ITOX,JDOY,JTOY)
      POLICY(I,J)= IJJ
   10 CONTINUE
C
C     DO 'ISTART' ITERATIONS BEFORE CHECKING LIST FOR TERMINATION.
C
      IF(K.LE.ISTART) GO TO 6
      DO 15 I=IDOX,ITOX,INCX
      DO 15 J=JDOY,JTOY,INCY
      IF( VALUE(I,J,INEW).NE.VALUE(I,J,IOLD) ) GO TO 6
   15 CONTINUE
C
C     RECURSION TERMINATED;  OUTPUT OF RESULTS FOLLOWS.
C
      WRITE(6,500) K,T(1),T(2)
      LINE=0
      DO 20 I=IDOX,ITOX,INCX
      DO 20 J=JDOY,JTOY,INCY
      IF( (I.EQ.T(1)).AND.(J.EQ.T(2)) ) GO TO 19
      LINE=LINE+1
      IF(LINE.GT.52) WRITE(6,500) K,T(1),T(2)
      IF(LINE.GT.52) LINE=1
      IJOUT=POLICY(I,J)
      IP=I+GOTOX(IJOUT)
      JP=J+GOTOY(IJOUT)
      WRITE(6,510) I,J, IP,JP, VALUE(I,J,INEW)
      GO TO 20
   19 WRITE(6,520) I,J
   20 CONTINUE
      GO TO 3
   60 CALL EXIT
  500 FORMAT('1',4X,'NUMBER OF ITERATIONS REQUIRED TO REACH SOLUTION:',
```

```
     1I6,//,5X,'FROM POINT     GO TO POINT     MIN COST TO TERMINAL(X=',
     2 I3,', Y=',I3,')',/,
     3  7X,'X     Y',9X,'X     Y',/,1X,130('*'),/)
  510 FORMAT(2X,2I6,5X,2I5, 5X,F20.2)
  515 FORMAT(16I5)
  520 FORMAT(2X,2I6,5X,'STOP--TERMINAL POINT.')
      RETURN
      END
      SUBROUTINE FIX(INEW,IOLD,K)
C
C     SUBROUTINE GIVES ALTERNATING VALUES OF '1' OR '2' TO INEW AND IOLD
C     DEPENDING ON  WHETHER K IS EVEN OR ODD:  K EVEN GIVES INEW VALUE
C     OF '2'.
C
    5 A=K/2.0
      IA=A
      A=A-IA
C
C     IF A IS ZERO K IS EVEN
C
      IF(A.EQ.0) GO TO 10
      INEW=1
      IOLD=2
      RETURN
   10 INEW=2
      IOLD=1
      RETURN
      END
      FUNCTION AMIN(IDOX,ITOX,JDOY,JTOY)
      COMMON /ARRAYS/ COST(100,100,8),VALUE(100,100,2),T(2)
      COMMON /VALUES/ NTO1X,NTO1Y,NTOX,NTOY,NDOX,NDOY,INCX,INCY,IOLD,
     1I,J,IJJ,NUMX,NUMY,ICOST,IBLOCK,NDEC,IGRIDX,IGRIDY,ZERO,C,G,H,
     2ISTART,ALPHA,P,BETA,IR,INTER
      INTEGER T
C
C     SUBROUTINE FINDS A RELATIVE MINIMUM VALVE OUT OF EACH POINT AND
C     ASSIGNS THE VALUE TO AMIN.  THE PATH POINT IS ASSIGNED TO IJJ.
C
      A=10**25
      DO 5 IT=1,8
      IP=I+GOTOX(IT)
      JP=J+GOTOY(IT)
      IF((IP.LT.IDOX).OR.(IP.GT.ITOX)) GO TO 5
      IF((JP.LT.JDOY).OR.(JP.GT.JTOY)) GO TO 5
      IF( VALUE(IP,JP,IOLD) + COST(I,J,IT) .LT.A) GO TO 4
      GO TO 5
    4 A=VALUE(IP,JP,IOLD) + COST(I,J,IT)
      IJJ=IT
    5 CONTINUE
      AMIN=A
      RETURN
      END
      SUBROUTINE PAGE(LINE)
      COMMON /ARRAYS/ COST(100,100,8),VALUE(100,100,2),T(2)
      COMMON /VALUES/ NTO1X,NTO1Y,NTOX,NTOY,NDOX,NDOY,INCX,INCY,IOLD,
     1I,J,IJJ,NUMX,NUMY,ICOST,IBLOCK,NDEC,IGRIDX,IGRIDY,ZERO,C,G,H,
     2ISTART,ALPHA,P,BETA,IR,INTER
C
C     SUBROUTINE PRODUCES A PAGE HEADING WHEN OUT PUTTING COSTS RESULTS.
```

```
C
      INTEGER T
      LINE=0
      WRITE(6,500) INCX,INCX,INCY,INCY,INCX,INCY,INCX,INCX,INCY,INCY,
     1INCX,INCY
      RETURN
  500 FORMAT('1',40X,'**COSTS OF TRAVELING FROM X,Y TO ADJOINING POINTS*
     1*',//,30X,'POINT 1    POINT 2    POINT 3    POINT 4    POINT 5
     2POINT 6    POINT 7    POINT 8',/,
     34X,'X    Y ALTITUDE  COSTS: X+',I2,',  Y : X+',I2,',Y+',I2,':   X
     4,Y+',I2,': X-',I2,',Y+',I2,'; X-',I2,',  Y : X-',I2,',Y-',I2,';
     5X ,Y-',I2,'; X+',I2,',Y-',I2,'.',/,1X,130('*') )
      END
      FUNCTION GOTO(DUMMY)
      COMMON /VALUES/ NTO1X,NTO1Y,NTOX,NTOY,NDOX,NDOY,INCX,INCY,IOLD,
     1I,J,IJJ,NUMX,NUMY,ICOST,IBLOCK,NDEC,IGRIDX,IGRIDY,ZERO,C,G,H,
     2ISTART,ALPHA,P,BETA,IR,INTER
C
C     SUBROUTINE IS USED FOR FINDING DIRECTION OF THE OUT POINTS.
C
      GOTO=0
      RETURN
C
C     OUT POINT IN THE X DIRECTION
C
      ENTRY GOTOX(IJOUT)
      GOTOX=-INCX
      IF((IJOUT.LT.3).OR.(IJOUT.EQ.8)) GOTOX=INCX
      IF((IJOUT.EQ.3).OR.(IJOUT.EQ.7)) GOTOX=0
      RETURN
C
C     OUT POINT IN THE Y DIRECTION
C
      ENTRY GOTOY(IJOUT)
      GOTOY=-INCY
      IF((IJOUT.EQ.1).OR.(IJOUT.EQ.5)) GOTOY=0
      IF((IJOUT.GT.1).AND.(IJOUT.LT.5)) GOTOY=INCY
      RETURN
      END
/*
//
```

REFERENCES

1. Shepard, Donald, "A Two Dimensional Interpolation Function for Irregularly Spaced Data," *Proceedings--1968 ACM National Conference*, pp. 517-523.

2. Fletcher, John G., et al., "Energy Costs," *Bioastronautics Data Book*, Paul Webb, M.D. (ed.), National Aeronautics and Space Administration, Washington, D. C., 1964.

3. Waddell, Gordon, "Energy Requirements of Mountain Walking," *Summit Magazine*, Vol. 12, No. 2, March 1966, p. 20.

4. Dreyfus, S. E., *An Appraisal of Some Shortest-Path Algorithms*, The Rand Corporation, RM-5433-1, September 1968.